PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-306849
(43)Date of publication of application : 21.11.1995

(51)Int.Cl. G06F 17/14

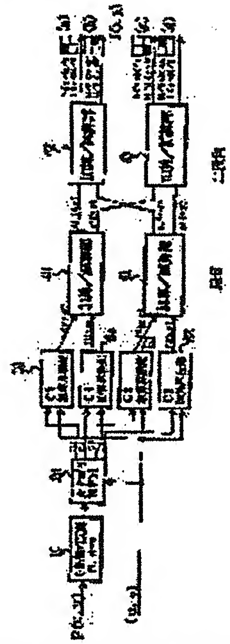
(21)Application number : 06-167741
(22)Date of filing : 20.07.1994

(71)Applicant : FUJITSU LTD
(72)Inventor : NAKAGAWA AKIRA
KAWAKATSU YASUHIRO
MORIMATSU EIJI
MATSUDA KIICHI

(30)Priority
Priority number : 06 74121 Priority date : 18.03.1994 Priority country : JP

(54) TWO-DIMENSIONAL INVERSE DISCRETE COSINE TRANSFORMATION SYSTEM

(57)Abstract:
PURPOSE: To provide the two-dimensional inverse discrete cosine transformation system which can increase the speed of an encoding process with small memory capacity.
CONSTITUTION: According to whether the horizontal and vertical frequencies of a base function that coefficients which are not 0 among DCT coefficients of one object block for the two-dimensional DCT of blocks of M×M pixels where M is power of 2 are an even function component or odd function component, components are classified into an even/even, an even/odd, an odd/even, and an odd/odd set, and values in a pixel area of (1/2).MI×(1/2).M which correspond to the positions and sizes of coefficients in each set and are put in a table after symmetry is removed are added and cumulated; after the cumulative addition, the cumulative values by the sets are (1/2).M×(1/2).M of four kinds of sets are added and subtracted by areas obtained by dividing a pixel area of size M×M for finding a value finally into four.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平7－306849

(43)公開日　平成7年(1995)11月21日

(51)Int.Cl.6	識別記号	庁内整理番号	F I	技術表示箇所
G06F 17/14				
			G06F 15/332	S

審査請求　未請求　請求項の数19　ＯＬ　（全 33 頁）

(21)出願番号　特願平6－167741

(22)出願日　平成6年(1994)7月20日

(31)優先権主張番号　特願平6－74121

(32)優先日　平6(1994)3月18日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(72)発明者　中川　章
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　川勝　保博
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　森松　映史
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　林　恒徳

最終頁に続く

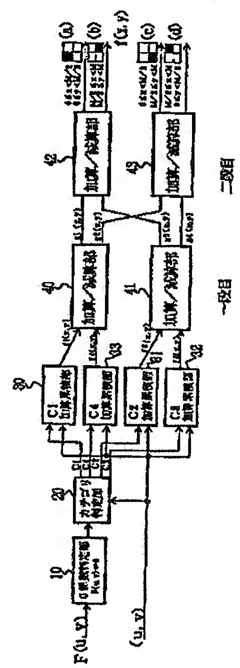
(54)【発明の名称】　二次元逆離散コサイン変換方式

(57)【要約】

【目的】小さいメモリ容量で復号化処理速度を高速化することができる二次元逆離散コサイン変換方式を提供する。

【構成】Mが２の巾乗であるM×Mの画素のブロックに対する２次元逆ＤＣＴを行うために対象とする１つのブロックのＤＣＴ係数のうち、値が０でない係数が表す基底函数の水平、垂直周波数が、偶函数成分、奇函数成分かに基づき、偶／偶、偶／奇、奇／偶、奇／奇の集合に分類し、各集合において、係数の位置と大きさに対応する、予めテーブル化しておいた、対象性を省いた（１／２）・M×（１／２）・Mの画素領域の値を加算累積し、累積加算完了後、各集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を４分割した各領域毎に、４種類の集合の（１／２）・M×（１／２）・Mの大きさの累積値を加算／減算する。

本発明の第一実施例ブロック図



【特許請求の範囲】

【請求項1】Mが2の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、
二次元逆離散コサイン変換を行うべき1つの周波数領域のM×MのブロックのＤＣＴ係数のうち、該係数の値が0でないものを検出する手段と、
該検出手段により検出された係数が表す二次元の基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かに基づき、（偶、偶）、（偶、奇）、（奇、偶）、（奇、奇）の4種類の集合に分類する手段と、
該分類手段により分類された各々の集合において、係数の位置と大きさに対応する、予めテーブル化しておいた、対象性を省いた 1/2M× 1/2Mの画素領域の値を加算累積する手段と、
更に、全ての0でない係数の加算累積完了後、各集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を4分割した各 1/2M× 1/2M領域毎に、4種類の集合の 1/2M× 1/2Mの大きさの累積値を加算／減算する手段を有して構成されることを特徴とする二次元逆離散コサイン変換方式。

【請求項2】請求項1において、
前記領域を4分割した各 1/2M× 1/2M領域毎に、4種類の集合の 1/2M× 1/2Mの大きさの累積値の加算・減算を行う手段は、各集合もしくは加算段階で求まった中間集合に属する係数が存在するかしないかを判断し、その結果に基づいて、その上位の加算段階で、その集合との加減算を適応的になくするように構成されることを特徴とする二次元逆離散コサイン変換方式。

【請求項3】請求項1において、
前記分類手段により分類される集合は、周波数領域のM×Mの係数の集合を、水平方向、垂直方向とも次数Mの偶の属性を持つ集合と定義し、
ついで、集合の求めかたに関し、それまでに分割された集合の基底が水平周波数、もしくは垂直周波数、もしくはその両方の属性が偶であった場合、その偶であった成分に関し、更に対称性を利用し、基底函数が偶函数である係数の集合と奇函数である係数の集合に分割し、
また、該分割の際に基底函数が偶函数であるものの分割された方向の属性を偶函数、奇函数であるものの集合の属性を奇とし、
該分割された各集合の分割された方向の次数をそれぞれもとの集合の1/2 とし、
更に、該分割された各集合に対して、適宜次数が1となるまで該分割を再帰的に行い、該各集合の（水平方向の次数）×（垂直方向の次数）を該集合の画素領域のサイズとするように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項4】請求項3において、前記加算累積手段は、水平方向、垂直方向とも、前記分割された各集合の（水平方向の次数）×（垂直方向の次数）のサイズの該集合に属する係数の位置と大きさに対応する種類の、画素領域のテーブルを有するように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項5】請求項1乃至4において、前記加算累積手段は、
各集合において、正の係数に対するテーブルのみを有し、負の係数に対しては、該係数の絶対値を用いて、該テーブルから値を求め、該求められた値を累積値から減算するようにしたことを特徴とする二次元逆離散コサイン変換方式。

【請求項6】請求項1乃至4において、前記加算累積手段は、
各集合もしくは加算段階で求めた中間集合に属する係数が存在するか否かを判断し、その結果に基づいて、それより上位の加算段階で、該集合との加減算を無くするようにして、演算量を削減することを特徴とする二次元逆離散コサイン変換方式。

【請求項7】Mが2の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、
二次元逆離散コサイン変換すべき1つのブロックの二次元離散コサイン変換係数のうち、該変換係数の値が0でないものを判定する手段と、
該変換係数が表す基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かで（偶、偶）、（偶、奇）、（奇、偶）、（奇、奇）の4種類の集合に分類する手段と、
係数が分類された集合ごとに、該集合に属する係数の位置と大きさに対する、対象性を省いた1/2 M×1/2 Mの画素領域の値を乗算により求める手段と、
該係数が分類された集合ごとに、該1/2 M×1/2 Mの画素領域の値を加算して累積する手段と、
全ての0でない係数の累算加算完了後、該4つの集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を4分割した各1/2 M×1/2 M領域毎に、4種類の集合の1/2 M×1/2 Mの大きさの累積値の加算・減算を行う手段を有して構成されることを特徴とする二次元逆離散コサイン変換を行う方式。

【請求項8】請求項7において、
前記対象性を省いた1/2 M×1/2 Mの画素領域の値を乗算により求める手段は、その集合に属する係数ごとに、対応する周波数の係数の大きさ1に対する1/2M×1/2 M画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずることにより求めることを特徴とする二次元逆離散コサイン変換を行う方式。

【請求項9】請求項7において、
前記対象性を省いた1/2 M×1/2 Mの画素領域の値を乗

算により求める手段は、1/2 M×1/2 Mの各画素ごとに、加算する画素において、該係数が表す縦周波数成分と該係数が表す横周波数成分と、該係数の大きさを乗ずることにより求めることを特徴とする二次元逆離散コサイン変換方式。

【請求項１０】請求項７において、

前記領域を４分割した各1/2 M×1/2 M領域毎に、４種類の集合の1/2 M×1/2 Mの大きさの累積値の加算・減算を行う手段は、各集合もしくは加算段階で求まった中間集合に属する係数が存在するかしないかを判断し、その結果に基づいて、その上位の加算段階で、その集合との加減算を適応的になくするようにしたことを特徴とする二次元逆離散コサイン変換を行う方式。

【請求項１１】Mが２の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、

二次元逆離散コサイン変換すべき１つのブロックの二次元離散コサイン変換係数のうち、その値が０でないものを判定する手段と、

該係数が表す基底函数の水平周波数、垂直周波数により、予め決められた種類Nの集合に、係数を分類する手段と、

集合ごとに、係数の位置と大きさに対する、対象性を省いた該係数の属する集合ごとに決められたサイズの画素領域の値を乗算により求める手段と、

集合ごとに、該画素領域の値を加算して累積する手段と、

全ての０でない係数の累積加算完了後、該集合ごとの画素領域の累積値を階層的に加減算を行って画素領域の中間集合を順次求め、かつ、該中間集合も階層的に加減算を行い、最終的にM×Mの大きさの画素領域の値を求める演算を行う手段を有することを特徴とする二次元逆離散コサイン変換方式。

【請求項１２】請求項１１において、

前記係数を分類するべき、予め決められた種類Nの集合は、第一にもとのM×Mの係数の集合を、水平方向、垂直方向とも次数Mの偶の属性を持つ集合と定義し、　次いで、集合の求めかたに関し、それまでに分割された集合の基底が、水平周波数、もしくは垂直周波数、もしくはその両方の属性が偶であった場合、その偶であった成分に関してその更なる対称性を利用し、基底函数が偶函数である係数の集合と奇函数である係数の集合に分割し、また、その際に基底函数が偶函数であるものの集合の分割した方向の属性を偶、奇函数であるものの属性を奇とし、

該分割された各集合の分割された方向の次数をそれぞれもとの集合の１／２とし、更に該分割された各集合に対して、次数が１以上であるかぎり、必要に応じて、該分割を再帰的に行い、該各集合の（水平方向の次数）×（垂直方向の次数）を、該集合の画素領域のサイズとするように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項１３】請求項１１において、

前記対象性を省いた、前記係数の属する集合ごとにきめられた画素領域の値を乗算により求める手段は、該集合に属する係数ごとに、該集合の（水平方向の次数）×（垂直方向の次数）のサイズの、係数の大きさ１に対する画素領域のパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずることにより求めるものであることを特徴とする二次元逆離散コサイン変換方式。

【請求項１４】請求項１１において、

前記対象性を省いた、前記係数の属する集合ごとにきめられた画素領域の値を乗算により求める手段は、該集合に属する係数ごとに縦方向と横方向のパターンを持ち、加算する画素において該係数が表す、該集合の（垂直方向の次数）のサイズの縦周波数成分と該集合の（水平方向の次数）のサイズの横周波数数成分と、該係数の大きさを乗ずることにより求めることを特徴とする二次元逆離散コサイン変換方式。

【請求項１５】請求項１１において、

前記全ての０でない係数の累積加算完了後、該集合毎の累積値を、階層的に加算・減算してM×Mの画素領域の変換値を求める手段は、各集合もしくは加算段階で求まった中間集合に属する係数が存在するかしないかを判断し、その結果に基づいて、その上位の加算段階で、その集合との加減算を適応的になくすようにしたことを特徴とする二次元逆離散コサイン変換方式。

【請求項１６】Mが２の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、

１つのブロック内の非０の係数の数をカウントし、非０の係数の数である有効係数の数を出力する手段と、

複数の種類の逆離散コサイン変換を行う手段と、

該複数の逆離散コサイン変換を行う手段を選択する手段を有し、

該選択する手段は、該有効係数の数を出力する手段からの有効係数の数により、複数の逆離散コサイン変換を行う手段のいずれかを選択するように構成されたことを特徴とする二次元逆離散コサイン変換方式。

【請求項１７】請求項１５において、

前記複数の逆離散コサイン変換のすくなくとも一つは、有効係数の個数に応じて、逆離散コサイン変換の演算量が変化するものであることを特徴とする二次元逆離散コサイン変換方式。

【請求項１８】請求項１５において、

前記複数の逆離散コサイン変換の少なくとも一つは、有効係数の個数に関わらず、一定の演算量で逆離散コサイン変換演算を行うものであることを特徴とする二次元逆離散コサイン変換方式。

【請求項19】請求項15において、
前記選択する手段は、前記入力された有効係数の個数において、もっとも演算回数が少ない逆離散コサイン変換を行う手段を選択すること特徴とする二次元逆離散コサイン変換方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、画像データの復号化側におけるテーブル参照（ルックアップ）方式を用いる二次元逆離散コサイン変換方式に関する。

【0002】

【従来の技術】大量のデータ、例えば画像データの伝送・蓄積を考える場合、効率において情報量を削減することが不可欠である。そのようなことから、データの相関性を取り除く方式として、様々な方式が検討されてきた。

【0003】その中でも二次元離散コサイン変換（以下、必要な場合、二次元ＤＣＴと略称する）およびその逆変換は、それぞれ符号化側、復号化側で画像データに対して用いることにより、画像の相関性を取り除くことができ、直交変換の中でも特に特性が良いことが知られている。

【0004】この様な点から、この二次元離散コサイン変換は、静止画の圧縮の標準化方式であるＪＰＥＧ、及び動画像符号化の標準化方式であるＭＰＥＧ、およびＨ．２６１に採用されている。

【0005】しかしながら、二次元離散コサイン変換、及びその逆変換は直交変換であり、実現するには行列演算が必要となる。そのため、演算量が非常に大きく、画像のＣＯＤＥＣ（符号化／復号化）をハード又はソフトで実現する場合のいずれによっても総演算量の大きな部分を占めている。

【0006】この演算量を削減するため、これまで、ハードウェアで実現する場合には回路規模の小さな方式、ソフトウェアで実現する場合には種々の高速アルゴリズムが検討の対象とされてきた。

【0007】ここで、以下は画像の復号化側で必要となる逆離散コサイン変換に限って議論を進める。従来の高速二次元逆離散コサイン変換（以下、必要な場合、二次元逆ＤＣＴと略称する）変換方式の代表例として、Ｃｈｅｎ等によるアルゴリズムがある（IEEE TRANSACTION ON COMMUNICATION, VOL. COM-25, No. 9, SEPTEMBER, 1977）。

【0008】Ｃｈｅｎのアルゴリズムによれば、８×８の二次元逆ＤＣＴを加算４１６回、乗算２５６回で行うことができる。しかしながら、マイクロプロセッサ、ＤＳＰ等で逆ＤＣＴ演算を行う場合には十分な演算速度が得られているとは言えない。特に動画データの復号化に用いた場合は、十分なフレームレートが得られず、動画としての使用には耐えない。

【0009】ところで、Ｈ．２６１、ＭＰＥＧ、ＪＰＥＧを用いた画像符号化標準方式において、伝送されるＤＣＴ係数には、以下の３つの性質がある。

【0010】（1）実用上十分な画質を得られる圧縮率で符号化した場合でも、伝送されるＤＣＴ係数のうち、せいぜい１／８が０でない有効な係数であり、残り７／８の係数は０である。

【0011】（2）０付近の係数の出現確率が非常に高い。

【0012】（3）係数は量子化されて伝送されるため、ＤＣＴ係数の全ての位置に対して値が存在するわけでなく、比較的少ない種類の係数のみが出現する。

【0013】ここで二次元逆ＤＣＴの定義式を数１のように考える。

【0014】

【数１】

$$f(x,y)=\sum_{u=0}^{M}\sum_{v=0}^{M}1/4C(u)C(v)F(u,v)\cos\left(\frac{(2x+1)u}{2M}\pi\right)\cos\left(\frac{(2y+1)v}{2M}\pi\right)$$

但し、$C(U)=1\ (u\neq 0)$、$=1/2^{1/2}\ (u=0)$
（定義によってC(U)の値が定数倍だけ異なることもある。）

【0015】上記数１において、C(u)、C(v)は、水平周波数成分及び垂直周波数成分に対するスケールファクタであり、F(u,v)は、二次元ＤＣＴされた係数である。Ｍは、２の巾乗である。また、ｘ、ｙは、二次元逆ＤＣＴされた画像データの二次元座標である。

【0016】更に、数１について見ると、逆ＤＣＴ変換結果ｆ(x,y)を求める時、Ｆ(u,v)＝０である水平周波数及び垂直周波数（ｕ，ｖ）の組み合わせについては、Σの内側の数２は０であり、何らの演算も行う必要がないことが分かる。

【0017】

【数２】

$$f'(x,y,u,v)=1/4C(u)C(v)F(u,v)\cos\left(\frac{(2x+1)u}{2M}\pi\right)\cos\left(\frac{(2y+1)v}{2M}\pi\right)$$

【0018】従って、Ｆ(u,v)≠０の条件を満たす水平周波数及び垂直周波数（ｕ，ｖ）についてのみ数２の演算を各ｘ，ｙに対して行い、その結果を各ｘ，ｙ毎に加算を行えばよい。

【0019】この結果、ｆ(x,y)は、次の数３のようになる。

【0020】

【数３】

$$f(x,y)=\sum_{(u,v),F(u,v)\neq 0} f'(F(u,v),x,y,u,v)$$

【0021】ここで、f'(level, x, y, u, v)〔level は係数F(u, v)のとりうる値〕を予めテーブルとして持っておけば、（x，y）座標の画素に対してF(u, v) ≠0の条件を満たす（u，v）についてのみ加算を行えば良い。

【0022】更に、M×Mのブロックに対して、m個の有効係数（0でない係数）が存在する場合、加算はmが2以上で生ずるので加算の総回数は、（m－1）×M^2である。

【0023】このことを、一般的な画像符号化方式に当てはめて検討してみる。MPEGやJPEGでは、一般に1ブロックのサイズは8画素×8画素（M＝8）が用いられており、これらの方式を用いて一般的な画像を復号した場合、各ブロックの0でない係数、つまり、F(u, v) ≠0　を満たす係数の数は、平均して8前後であることが知られている。

【0024】この事より、ある座標の画素の値を求めるのに、せいぜい8回の加算のみで値が求まることが分かる。8×8のブロック全体で考えても、448回の加算である。

【0025】一方、従来の高速アルゴリズムの一つであるChenのアルゴリズムでは、8×8の1ブロックあたり加算416回、乗算256回を要する。仮に加算と乗算が同じ速度で処理できるとしても加算換算672回の演算を要する。一般に乗算は、加算に比べて非常に処理に時間を要する。

【0026】更に、M×Mの二次元逆DCTを行うと、MPEGなどで採用されているレベルがLビットの場合、必要なワード数は、

（uの種類）×（vの種類）×（xの種類）×（yの種類）×（レベルの種類）＝M^4 ×2^L （ワード）

となる。

【0027】これは、現在、MPEGなどで一般に用いられているL＝12、M＝8の場合、16Mワードとなる。1ワードを2バイトとすると、必要なテーブルの容量は32Mバイトとなる。このような大きなテーブルを持つには、ハード、ソフトのいずれでも現行技術では実行するのには困難が伴う。

【0028】従って、本出願人は先に、持つべき係数の（u，v）の種類を制限して、逆DCTテーブルサイズの削減を行い、受信した係数に対するテーブルがあればその値を用い、係数に対するテーブルがなければ数2に基づいて値を計算する方式を提案している。

【0029】そして、この方式に基づき逆変換を行う構成の概念ブロック図は、図22に示される。更に、図23はその動作フローチャートである。

【0030】図22において、0係数判別部141によって、入力される基底函数F(u, v)について、0であるか否かを判定し、F(u, v) ≠0であるF(u, v) のみを出力する（ステップS151：図23）。

【0031】ついで、テーブルミス判定部142において、出力されるF(u, v) について、f'(level, x, y, u, v)が予めテーブルとして登録されているか否かを判断し、その結果スイッチSW1及びSW1' を制御する（ステップS152）。

【0032】即ち、テーブルミス判定部142において、0係数判別部141から出力されるF(u, v) について、f'(level, x, y, u, v)が予めテーブルとして登録されている場合は、スイッチSW1及びSW1' を逆DCTテーブル143に接続する。

【0033】したがって、このテーブルから対応する画素領域値f'(x, y)が出力される（ステップS153）。またf'(level, x, y, u, v)が予めテーブルとして登録されていない場合は、スイッチSW1及びSW1' を逆DCT演算部144に接続し、画素領域値f'(x, y)を計算して出力する（ステップS154）。

【0034】このようにして出力される画素領域値f'(X, Y)が累積加算部145において累積される（ステップS155）。この処理を1ブロック終了するまで継続する（ステップS156）。

【0035】しかしながら、このような本出願人が先に提案する方式においては、テーブルにない係数が来た場合、毎回数2の計算を逆DCT演算部144において行わねばならず、画像の種類によっては、演算量が大幅に増加してしまう虞がある。

【0036】

【発明が解決しようとする課題】したがって、本発明の基本的な目的は、小さいメモリ容量で復号化処理速度を高速化することができる画像データ処理における二次元逆離散コサイン変換方式を提供することにある。

【0037】更に本発明の目的は、逆DCTすべき1つのブロックのDCT係数のうち、その値が0でないものをその係数が表す基底函数の水平、垂直周波数がそれぞれ偶函数か奇函数かで集合に分類し、それぞれの集合において、係数の位置と大きさに対応する値をテーブル化して用いるようにする二次元逆離散コサイン変換方式を提供するにある。

【0038】更にまた本発明の目的は、上記周波数成分に対応するテーブルの作成において、その周波数に対し、基本もしくは代表となる係数の値のみテーブルを作成するようにする二次元逆離散コサイン変換方式を提供することにある。

【0039】本発明の更なる目的は、以下の説明及び添付された請求の範囲の記載から明らかになる。

【0040】

【課題を解決するための手段】本発明にしたがう二次元逆離散コサイン変換方式の基本構成は、Mが2の巾乗で

あるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、二次元逆離散コサイン変換を行うべき１つの周波数領域のM×MのブロックのＤＣＴ係数のうち、前記係数の値が０でないものを検出する手段と、前記検出手段により検出された係数が表す二次元の基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かに基づき、（偶、偶）、（偶、奇）、（奇、偶）、（奇、奇）の４種類の集合に分類する手段と、前記分類手段により分類された各々の集合において、係数の位置と大きさに対応する、予めテーブル化しておいた、対象性を省いた 1/2M× 1/2Mの画素領域の値を加算累積する手段と、更に、全ての０でない係数の加算累積完了後、各集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を４分割した各1/2M× 1/2M領域毎に、４種類の集合の 1/2M× 1/2Mの大きさの累積値を加算／減算する手段を有して構成される。

【００４１】更に、本発明の上記基本構成と対比される別の構成は、Mが２の巾乗であるM×Mで構成される画素領域のブロックを求める二次元逆離散コサイン変換を行う方式であって、二次元逆離散コサイン変換すべき１つのブロックの二次元離散コサイン変換係数のうち、該変換係数の値が０でないものを判定する手段と、変換係数が表す基底函数の水平周波数、垂直周波数が、それぞれ偶函数成分、奇函数成分かで（偶、偶）、（偶、奇）、（奇、偶）、（奇、奇）の４種類の集合に分類する手段と、係数が分類された集合ごとに、この集合に属する係数の位置と大きさに対する、対象性を省いた1/2 M×1/2 Mの画素領域の値を乗算により求める手段と、該係数が分類された集合ごとに、1/2 M×1/2 Mの画素領域の値を加算して累積する手段と、全ての０でない係数の累算加算完了後、該４つの集合ごとの累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を４分割した各1/2 M×1/2 M領域毎に、４種類の集合の1/2 M×1/2 Mの大きさの累積値の加算・減算を行う手段を有して構成される。

【００４２】この別の構成における態様として、上記対象性を省いた1/2 M×1/2 Mの画素領域の値を乗算により求める手段は、その集合に属する係数ごとに、対応する周波数の係数の大きさ１に対する1/2 M×1/2 M画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずるように構成される。

【００４３】

【作用】本発明は、基本的構成において、ＤＣＴ係数を分類し、その分類された集合ごとに累積加算し、最後に各集合毎の加減算により逆ＤＣＴを行う。

【００４４】その際、それぞれの集合において、係数の位置と大きさに対応する対象性を省いた 1/2M× 1/2Mの画素領域の値が予めテーブル化されている。そして、０でない係数に対応する値をテーブルから読みだし、加算累積する。

【００４５】ついで、全ての０でない係数の累積完了後、各集合毎の累積値を、最終的に値を求めるべきM×Mの大きさの画素領域を４分割した各 1/2M× 1/2M領域毎に、４種類の集合の 1/2M× 1/2Mの大きさの累積値を加算／減算するよう構成される。

【００４６】一方、上記別の構成においては、その集合に属する係数ごとに、集合に属する係数の位置と大きさに対する、対象性を省いた1/2 M×1/2 Mの画素領域の値を乗算により求め、ついで前記係数が分類された集合ごとに、1/2 M×1/2 Mの画素領域の値を加算して累積するようにしている。

【００４７】より具体的には、上記別の構成では、その集合に属する係数ごとに、対応する周波数の係数の大きさ１に対する1/2 M×1/2 M画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを乗ずるようにしている。 したがって、かかる本発明の上記基本構成ならびに別の構成においては、少ないメモリ容量即ち、小さなテーブルサイズで逆ＤＣＴのための演算回数の削減と、演算処理の高速化が可能である。

【００４８】

【実施例】以下実施例の説明において、同一または、類似のものには同一の数字及び記号を付して説明する。

【００４９】図１は、Ｉピクチャ用のＭＰＥＧデコーダであり、本発明の対象とする二次元逆離散コサイン変換方式が適用される部分の位置づけを示すものである。

【００５０】先ず符号化された画像データのビットストリームは、可変長復号化回路１に入力され、対応するレベルに変換される。次いで量子化回路２において、二次元ＤＣＴ係数Ｆ(u,v) を出力する。

【００５１】二次元ＤＣＴ係数Ｆ(u,v) は、本発明方式に従う二次元逆ＤＣＴ回路３により、元の画像の画素データｆ(x,y) が再生される。

【００５２】図２は、本発明の方式を採用する二次元逆ＤＣＴ回路３の第一の実施例ブロック図である。ここで、図２の実施例の構成及び動作の説明に先立って、本発明の完全な理解のために数的考察を先に行う。

【００５３】一般に汎用的な画像符号化その他で用いられる離散コサイン変換の空間の値、及び基底の数は２の巾乗である。このようなことから、Mが２の巾乗である場合を考える。０≦ｘ，ｙ＜M／２において定義される以下のような数４を考える。

【００５４】

【数４】

$$f1(x,y)=\sum_{u'=0}^{M/2}\sum_{v'=0}^{M/2}\frac{1}{4}C(2u')C(2v')F(2u',2v')\cdot\cos\left(\frac{(2x+1)2u'}{2M}\pi\right)\cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$f2(x,y)=\sum_{u'=0}^{M/2}\sum_{v'=0}^{M/2}\frac{1}{4}C(2u'+1)C(2v')F(2u'+1,2v')\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$f3(x,y)=\sum_{u'=0}^{M/2}\sum_{v'=0}^{M/2}\frac{1}{4}C(2u')C(2v'+1)F(2u',2v'+1)\cdot\cos\left(\frac{(2x+1)2u'}{2M}\pi\right)\cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

$$f4(x,y)=\sum_{u'=0}^{M/2}\sum_{v'=0}^{M/2}\frac{1}{4}C(2u'+1)C(2v'+1)F(2u'+1,2v'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

【００５５】上記数４から、ｆ1 では、水平周波数ｕ（＝２ｕ’），垂直周波数ｖ（＝２ｖ’）ともに偶数（集合Ｃ１）
ｆ2 では、水平周波数ｕ（＝２ｕ’＋１）が奇数、垂直周波数ｖ（＝２ｖ’）が偶数（集合Ｃ２）
ｆ3 では、水平周波数ｕ（＝２ｕ’）が偶数、垂直周波数ｖ（＝２ｖ’＋１）が奇数（集合Ｃ３）
ｆ4 では、水平周波数ｕ（＝２ｕ’＋１），垂直周波数ｖ（＝２ｖ’＋１）ともに奇数（集合Ｃ４）の成分の函数に関する部分和であることが分かる。

【００５６】つまり、ｆ1 ～ｆ4 のそれぞれは、（ｕ，ｖ）の組に関して、高々M^2 ／４種類の係数に対する加算が行われる。また、それぞれの部分和の函数の（ｘ，ｙ）の画素数に関しても、各M^2 ／４　画素に対するものである。

【００５７】さらに、０≦ｘ，ｙ＜Ｍ／２において定義されるような、以下のような部分和を考える。

【００５８】

$$g1(x,y)=f1(x,y)+f4(x,y)\quad\cdots(1)$$

$$g2(x,y)=f1(x,y)-f4(x,y)\quad\cdots(2)$$

$$g3(x,y)=f2(x,y)+f3(x,y)\quad\cdots(3)$$

$$g4(x,y)=f2(x,y)-f3(x,y)\quad\cdots(4)$$

二次元離散コサイン変換を行った結果の各画素の値は、これら式（１）～（４）を用い、またＤＣＴ係数の対称性を利用して、以下の様にして求められる。この結果は下記(a) ～(d) に示した各領域において、数１と等価になる。

【００５９】(a) ０≦ｘ，ｙ＜Ｍ／２　なる（ｘ，ｙ）に対して、

$$f(x,y)=g1(x,y)+g3(x,y)\quad\cdots(5)$$

(b) Ｍ／２≦ｘ＜Ｍ，０≦ｙ≦Ｍ／２　なる（ｘ，ｙ）に対して、

$$f(x,y)=g2(M-x,y)-g4(M-x,y)\quad\cdots(6)$$

(c) ０≦ｘ＜Ｍ／２，Ｍ／２≦ｙ＜Ｍ　なる（ｘ，ｙ）に対して、

$$f(x,y)=g2(x,M-y)+g4(x,M-y)\quad\cdots(7)$$

(d) $M/2 \leq x,\ y < M$ なる（x，y）に対して、

$$f(x,y) = g1(M-x, M-y) - g3(M-x, M-y) \quad \cdots (8)$$

以上により、二次元逆離散コサイン変換が表される。但し、上記式（１）～（４）及び上記（５）～（８）に示した $f1$ ～ $f4$ から $f(x,y)$ を求める加算の順序は、あくまでも一例であり、他の加算順序も勿論本発明の範囲から除外されるものではない。

【００６０】次に、これらの式を用いた二次元逆離散コサイン変換の方法について、考察する。上記数４の各々の式（１）～（４）を見ると、これは、数１と良く似た形をしている事がわかる。

【００６１】このことから、それぞれ式（１）～（４）において、$F(u,v) \neq 0$ の条件を満たす（u，v）についてのみ数４の各々のΣの中の演算を、各ｘ，ｙに対して行い、その結果を各座標の $f(x,y)$ 毎に加算を行えばよい。これを式で表すと次の様になる。

【００６２】f1 に対しては、（u，v）$\subseteq$ C１の係数に対して、数５になる。

【００６３】

【数５】

$$f1(x,y) = \sum_{(u,v),\,F(u,v)\neq 0} f1'(F(u,v), x, y, u, v)$$

【００６４】f2 に対しては、（u，v）$\subseteq$ C２の係数に対して、数６になる。

【００６５】

【数６】

$$f2(x,y) = \sum_{(u,v),\,F(u,v)\neq 0} f2'(F(u,v), x, y, u, v)$$

【００６６】f3 に対しては、（u，v）$\subseteq$ C３の係数に対して、数７になる。

【００６７】

【数７】

$$f3(x,y) = \sum_{(u,v),\,F(u,v)\neq 0} f3'(F(u,v), x, y, u, v)$$

【００６８】f4 に対しては、（u，v）$\subseteq$ C４の係数に対して、数８になる。

【００６９】

【数８】

$$f4(x,y) = \sum_{(u,v),\,F(u,v)\neq 0} f4'(F(u,v), x, y, u, v)$$

【００７０】ここで、予め $f1'(level, x, y, u, v)$ ～ $f4'(level, x, y, u, v)$ を計算して、テーブルとして持っておけば、式ｆ１～ｆ４ に対しては、それぞれ集合Ｃ１～Ｃ４に属する係数が $F(u,v) \neq 0$ を満たすものについてのみ、加算を行えば良いことになる。

【００７１】式ｆ１～ｆ４が求まれば、上記式（１）～（４）の g1 ～g4 及び式（５）～（８）に従って、画素データである $f(x,y)$ が求まる。

【００７２】この手法を用いて、M×Mの二次元逆ＤＣＴを行う場合の必要なテーブルの数、および、演算量について更に、考察を行う。

【００７３】まず、数５～数８を、先に検討した、レベルがＬビットの場合を実現する例について考察する。ここでは、ｘ，ｙ，ｕ，ｖともに、$0 \leq x,\ y,\ u,\ v < M/2$ を満たしている。また、数５～数８に対して、それぞれ独立にテーブルが４種類必要である。これらのことより、数５を実現するために必要なテーブルの大きさは、

（テーブルの種類）×（ｕの種類）×（ｖの種類）×（ｘの種類）×（ｙの種類）×（レベルの種類）＝ $4 \times (M/2)^4 \times 2^L$（ワード）＝ $M^4 \times 2^{L-2}$（ワード）となる。

【００７４】したがって、この場合のテーブルの大きさは、既に検討した本出願人により先に提案している方式の１／４である。

【００７５】次に、演算量について考察する。先の仮定と同様に、$F(u,v) \neq 0$ を満たす係数の個数をｍとした場合について考える。ここで、０でないｍ個の係数を以下の様に分ける。

【００７６】Ｃ１に属する係数の個数：ｍ１

Ｃ２に属する係数の個数：ｍ２

Ｃ３に属する係数の個数：ｍ３

Ｃ４に属する係数の個数：ｍ４

$m1 + m2 + m3 + m4 = m$

数５～数８において、０でない係数が１つ存在した場合に必要な個数は、

（ｘの種類）×（ｙの種類）＝ $(M/2)^2$

であるから、数５～数８における必要な加算回数 I_1 は以下のようになる。

【００７７】$I_1 = (M/2)^2 \times (m1-1) + (M/2)^2 \times (m2-1) + (M/2)^2 \times (m3-1) + (M/2)^2 \times (m4-1)$

$= (M/2)^2 \times (m1+m2+m3+m4-4)$

$= 1/4\, M^2 \times m - M^2$　となる。

【００７８】また、前記式（１）～（４）の g1 ～g4 を求めるためには、必要な加算の回数 I_2 は、$I_2 = 4 \times (M/2)^2 = M^2$　となる。

【００７９】同様に、上記(a) ～(d) を求めるためにも、必要な加算の回数 I_3 は、

$I_3 = 4 \times (M/2)^2 = M^2$　となる。

【００８０】以上のことから、この演算に必要な総加算回数Ｉは、

$I = I_1 + I_2 + I_3 = M_2 + 1/4\, M^2 \times m$　となる。

【００８１】このことから、演算量についても、$m > 3$ において、既に説明した先に本出願人により提案した方式における演算量である $(m-1) \times M^2$ より削減される。

【００８２】上記した本発明の方式を採用する第一の実

施例ブロックが図２乃至図５に示され、その動作フローが図６及び図７に示される。

【００８３】以下図２乃至図５を、図６及び図７の動作フローに対応させながら説明する。

【００８４】図２において、１０は、０係数判定部であり、入力される基底函数の内、F(u,v) ≠０である即ち、０でない函数のみ出力する（ステップＳ１０１：図６）。

【００８５】２０は、F(u,v) ≠０である函数を分類して出力するカテゴリ判定部であり、その構成例は、図３に示される。即ち、水平及び垂直周波数（ｕ、ｖ）の奇数、偶数を判定して、集合Ｃ１～Ｃ４として分類し、集合毎に出力する（ステップＳ２０１～Ｓ２０３：図６）。

【００８６】図３の構成の実現は、マイクロコンピュータ等によりソフト的に実現可能である。ここで集合Ｃ１は、先に説明したようにｕ、ｖともに偶数の場合であり、集合Ｃ２は、ｕが奇数で、ｖが偶数の場合である。

【００８７】更に、集合Ｃ３は、ｕが偶数で、ｖが奇数の場合であり、集合Ｃ４は、ｕ、ｖともに奇数の場合である。

【００８８】これら各集合ごとの出力は、Ｃ１加算累積部３０乃至Ｃ４加算累積部３３の対応するものに入力され、累積される。Ｃ１加算累積部３０乃至Ｃ４加算累積部３３の各々は、同一構成であり、その一例が図４に示される。

【００８９】集合Ｃｎ（ｎ＝１、２、３、４）に属する係数F(u,v) と、水平周波数Ｕ、垂直周波数Ｖが入力される。３０１は、集合Ｃｎに属するF(u,v) に対応する画素領域の値が登録記憶されるメモリーのテーブルである。

【００９０】３０２は、加算回路であり、３０３は、累積値を記憶するメモリである。メモリ３０３の出力が加算回路３０２に帰還され、テーブル３０１の出力と再度加算されることにより累積される（ステップＳ３０１～Ｓ３０４：図6）。

【００９１】そして上記処理が１ブロック分終了するまで繰り返される（ステップＳ３０５）。

【００９２】図２に戻り、加算累積部３０乃至３３の出力は、第一段目の加算／減算部４０及び４１に入力される。加算／減算部４０には、Ｃ１加算累積部３０とＣ４加算累積部３３のそれぞれから f1 及び f4 が入力される。

【００９３】一方、加算／減算部４１には、Ｃ２加算累積部３１とＣ３加算累積部３２のそれぞれから f2 及び f3 が入力される。

【００９４】加算／減算部４０及び４１は、入力される f1 及び f4 、 f2 及び f3 に対し、それぞれ加算／減算を行い、g1 及び g2 、 g3 及び g4 を出力する（ステップＳ４０１）。かかる加算／減算は、先に式（１）～（４）の g1 乃至 g4 に関して説明した通りである。

【００９５】ここで加算／減算部４０、４１及び、後に説明する第二段目の加算／減算部４２、４３は、同一構成であり、その一例が図５に示される。即ち、第一の入力（Ｉｎｐ－１）と第二の入力（Ｉｎｐ－２）は、それぞれ加算回路４０１及び減算回路４０２に入力され、第一の入力（Ｉｎｐ－１）と第二の入力（Ｉｎｐ－２）との加算及び減算が行われる。

【００９６】加算結果として第一の出力（Ｏｕｔ－１）、減算結果として第二の出力（Ｏｕｔ－２）が得られる。

【００９７】更に、第一段目の加算／減算部４０及び４１の出力は、第二段目の加算／減算部４２、４３に入力され、同様に加算／減算される。即ち、加算／減算部４２では、第一段目の加算／減算部４０からの g1 と、加算／減算部４１からの g3 を加算／減算する。

【００９８】一方、加算／減算部４３では、第一段目の加算／減算部４０からの g2 と、加算／減算部４１からの g4 を加算／減算する。

【００９９】この結果、式（１）～（４）の g1 ～g4 に基づき、先に説明した通り、４つの領域（ａ）乃至（ｄ）に対応する、二次元逆ＤＣＴを行った結果 f(x,y)が得られる（ステップＳ４０２）。

【０１００】次に再帰的方法による本発明の第一の実施例の拡張について説明する。

【０１０１】先ず、ｕ、ｖともに偶数成分である、カテゴリＣ１に属する係数に関する演算である数４の f1(x,y) を書き直すと、数９のとおりになる。

【０１０２】

【数９】

$$f1(x,y) = \sum_{u'=0}^{M/2} \tfrac{1}{4}C(2u')\cdot\cos\left(\frac{(2x+1)2u'}{2M}\pi\right)\cdot \sum_{v'=0}^{M/2} F(2u',2v')\cdot C(2v')\cdot\cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$= \underbrace{\sum_{u'=0}^{M/2} \tfrac{1}{4}C(2u')\cdot\cos\left(\frac{(2x+1)u'}{2(M/2)}\pi\right)\cdot}_{①} \underbrace{\sum_{v'=0}^{M/2} F(2u',2v')\cdot C(2v')\cdot\cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right)}_{②}$$

【０１０３】ここで数９をよく見ると、部分①，②は、ともに次数がM／２の逆ＤＣＴの式になっていることが分かる。つまり、この数９は数１のMをM／２と変え，ｕ(＝２ｕ’）をｕ’に，ｖ（＝２ｖ’）をｖ’に置き換えただけのものと同じ式である。

【０１０４】したがって、このｆ1 に関しては，
ｆ11(x,y) では、ｕ’、ｖ’ともに偶数（集合C11）
ｆ12(x,y) では、ｕ’が奇数、ｖ’が偶数（集合C12）
ｆ13(x,y) では、ｕ’が偶数、ｖ’が奇数（集合C13）
ｆ14(x,y) では、ｕ’、ｖ’ともに奇数（集合C14）
の様に数１を、数４に分解して式（１）～（４）のｇ1～ｇ4 によりｆ(x,y) を求めたと全く同じ方法で、更にｆ1(x,y)に属する係数をｆ11～ｆ14のように４つの集合に分割し、ｆ1(x,y)を求めるための演算量，テーブルサイズを小さくすることが可能である。

【０１０５】また、ｆ11(x,y)、ｆ12(x,y)、ｆ13(x,y) は分割した基底の両方もしくは片方が偶数成分であるため、さらなる分割も可能である。

【０１０６】次に、ｕが奇数成分、ｖが偶数成分である、カテゴリＣ２に属する係数に関する演算である数４のｆ2(x, y) を書き直すと、数１０のとおりになる。

【０１０７】

【数１０】

$$f2(x,y) = \sum_{u'=0}^{M/2} \tfrac{1}{4}C(2u'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cdot \sum_{v'=0}^{M/2} F(2u',2v')\cdot C(2v')\cdot\cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

$$= \underbrace{\sum_{u'=0}^{M/2} \tfrac{1}{4}C(2u'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cdot}_{①} \underbrace{\sum_{v'=0}^{M/2} F(2u',2v')\cdot C(2v')\cdot\cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right)}_{②}$$

【０１０８】この数１０をみると、偶数成分であるｖに関しては、部分②の様に次数M／２の逆ＤＣＴの式となるが、奇数成分であるｕに関しては、部分①の様にカテゴリＣ１の場合と異なり、次数M／２の逆ＤＣＴの式とはならない。この場合には，数１０は、以下のように分解できる。

【０１０９】奇数のｖ’に対しては、数１１のとおりになる。

【０１１０】

【数１１】

$$h21(x,y)=\sum_{u'=0}^{M/2}\tfrac{1}{4}C(2u'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cdot\sum_{v'=0}^{M/4}F(2u',4v'')\cdot C(4v'')\cdot\cos\left(\frac{(2y+1)2v''}{2(M/2)}\pi\right)$$

【０１１１】偶数のｖ’に対しては、数１２のとおりになる。

【０１１２】

【数１２】

$$h22(x,y)=\sum_{u'=0}^{M/2}\tfrac{1}{4}C(2u'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cdot\sum_{v'=0}^{M/4}F(2u',2(2v''+1))\cdot C(2(2v''+1))\cdot\cos\left(\frac{(2y+1)(2v''+1)}{2(M/2)}\pi\right)$$

【０１１３】そして、(e)　０≦ｘ＜Ｍ／２、０≦ｙ＜Ｍ／４なる（ｘ，ｙ）に対して、

$$f2(x,y)=h21(x,y)+h22(x,y)\quad\cdots(9)$$

(f)　０≦ｘ＜Ｍ／２、Ｍ／４≦ｙ＜Ｍ／２なる（ｘ，ｙ）に対して、

$$f2(x,y)=h21(x,M/4-y)-h22(x,M/4-y)\quad\cdots(10)$$

なる式が表現できる。

【０１１４】このことから、この場合も、数５～数８と同様にテーブルを用いて$h21(x,y)$, $h22(x,y)$ を求め、これから上記式（９）、（１０）を用いて$f2(x,y)$を求めることができる。

【０１１５】この場合も、テーブルサイズ、演算量とも、数６を用いて求めるより削減されている。この場合、まず$h21(x,y)$, $h22(x,y)$ から$f2(x,y)$を求め、あとは、先に示した式（１）～（４）及び式（５）～（１０）に示した方法で$f(x,y)$ が求まる。

【０１１６】また、$h21(x,y)$ に関しては、ｖ”成分が偶数成分なので、さらなる分割が可能である。

【０１１７】以上に示した様に、その基底が偶函数の集合に対して、それらの基底は次数を半分にしたＤＣＴの基底と等価となるので、その集合のさらなる分割が可能となる。

【０１１８】この様に、基底が偶函数の成分に関しては、更にその係数の集合を分割でき、その結果、演算量とテーブルサイズともにより小さくできる。また、この分割は基底が偶数であるかぎり必要に応じて再帰的に行うことが可能である。

【０１１９】次に本発明の第二の実施例について説明する。

【０１２０】一般に、係数であるLevel Ｆ(u,v) は、正と負のどちらも存在する。また、Level が正負反転すると、f1’(level,x,y,u,v) ～f4’(level,x,y,u,v) も、正負反転する。このことから、f1’(level,x,y,u,v) ～f4’(level,x,y,u,v) に関しても、level が正の値、もしくは負の値に対する、どちらか一方のみ用意すればよい。

【０１２１】この場合において、f1 ～f4 を求める方法を以下に示す。

【０１２２】ｎ＝１、２、３、４として、fn は、（ｕ，ｖ）⊆Cn の係数に対して、ｆｎ（x, y)は、数１３のとおりである。

【０１２３】

【数１３】

$$fn(x,y)=\sum_{(u,v),\,F(u,v)>0}fn'(|F(u,v)|,x,y,u,v)-\sum_{(u,v),\,F(u,v)<0}fn'(|F(u,v)|,x,y,u,v)$$

【０１２４】この数１３を用いると、

1/2 ×（テーブルの種類）×(uの種類) ×(vの種類) ×(xの種類) ×(yの種類) ×（レベルの種類）＝１／２×４×(Ｍ／２)^4 ×2^L （ワード）＝M^4 ×2^{L-3} （ワード）

となる。

【０１２５】これは先に考察した演算量（M^4 ×2^{L-2} ）に対し、更に半減する。したがって、既に説明した、本出願人が先に提案した図２２、図２３に示す方式に比べ１／８となり、大幅にテーブルサイイズが削減できる。この方法により、更に必要とするテーブルの大きさを半分にする事が可能となる。

【０１２６】さらに、ＤＣＴ係数がＣ１～Ｃ４のうち、Ｃ１，Ｃ２，Ｃ４に属するもののみが存在し、Ｃ３に属

するものが存在しない場合を考える。この場合、f3(x, y)は0である。この場合、g3(x, y)、g4(x, y)は、単にf2 (x, y)をコピーすれば良いだけで、加算が必要なくなる。

【0127】C3に加えて、C2に属する係数も存在しない場合は、もはや、g3(x, y)、g4(x, y) も0で、その結果、g3、g4 を求める為の加算は必要でなく、さらに、f (x, y)も単に、g1(x, y)、g2(x, y) のコピーだけで値が求まる。

【0128】このように、予め、加算を行う際に、片方の加算すべき集合に属する係数が存在するかどうかを基に条件判断し、適応的に加算を行うことにより、さらに平均的な加算回数を削減することが可能となる。

【0129】この、適応的に加減算を行う方法、および、正負のテーブルをまとめてテーブルサイズを小さくする方法を用いた、本発明方式の第二実施例ブロック図を図8～図10に、その動作フローチャートを図11～図14に示す。

【0130】第一の実施例の場合と同様に実施例ブロック図を動作フローチャートと対象しながら説明する。

【0131】図8において、0係数判定部10及びカテゴリ判定部20の構成及び動作は、図2に示す場合と同様であり、0係数の判定（ステップS101：図11）及びカテゴリの判定が行われる（ステップS201乃至S203）。

【0132】カテゴリ判定部20の出力は、対応する累積加算部30乃至33に入力される。そこで累積加算される。第二の実施例における累積加算部30乃至33の構成例が図9に示される。図13は、累積加算部30乃至33の対応する累積加算動作フローである。

【0133】図9において、F(u,v) として有効係数が一度でも入力される場合は、スイッチ307により“1”にセットする（ステップS132）。即ちスイッチ307により、“1”が出力されるように制御する。

【0134】絶対値化部304によりカテゴリ判定部20の出力F(u,v) が、絶対値化される。ついで、入力されるF(u,v) が正であるか否かが判定される（ステップS132）。

【0135】テーブル305は、f'n(level, u, v, x, y)、但し Level＞0に対する画素領域の値を登録記憶しており、絶対値化されたF(u,v) により対応する画素領域の値が読みだされる。

【0136】306は、加算／減算回路であり、テーブル305の出力に対し、F(u,v) が正であれば、そのまま累積加算値を記憶するメモリ303の出力と加算し、負で有れば、極性を反転してメモリ303の出力と加算する（ステップS133、S134）。

【0137】これをx, yのループの終わりまで継続する（ステップS135、S136）。したがって、テーブル305は、F(u,v) により対応する画素領域の値として、正負ある画素領域の値の内、正の画素領域の値のみ記憶すればよくメモリ容量が削減出来る。

【0138】そして、上記の処理は1ブロック終了するまで継続される（ステップS305）。

【0139】各累積加算部30乃至33から有効係数の有無により出力されるe1 ～e4 は、一段目の適応的加算／減算部50、51及びオアゲート回路60、61に入力される。

【0140】即ち、累積加算部30及び33からの出力e1 及びe4 は、適応的加算／減算部50及びオアゲート回路60に入力し、累積加算部31及び32からの出力e2 及びe3 は、適応的加算／減算部51及びオアゲート回路61に入力する。

【0141】一段目の適応的加算／減算部50、51及び、それらの出力及びオアゲート60、61の出力が入力される二段目の適応的加算／減算部52、53の構成例は、図10に示される。即ち、第一及び第二の入力Inp－1、Inp－2及び入力S1 とS2 の組み合わせに対応して、加算又は減算を行うように構成される演算回路である。また、図14にその動作フローが示される。

【0142】図14において、en 、em をそれぞれS1 、S2 とすると、入力S1 とS2がともに“1”である場合（ステップS141：y，S142：y）、第一の出力として第一及び第二の入力Inp－1、Inp－2の和が、第二の出力として第一及び第二の入力Inp－1、Inp－2の差が演算され、出力される（ステップS144）。

【0143】更に、入力S1 が“1”、入力S2が“0”である場合（ステップS141：y，S142：n）、第一及び第二の出力ともに第一の入力Inp－1が出力される（ステップS145）。

【0144】また、入力S1 が“0”であり、入力S2 が“1”である場合（ステップS141：n，ステップS143：y）、第一の出力にはInp－2、第二の出力にはInp－2に（－1）乗じたものが出力される（ステップS146）。更に、入力S1 とS2 がともに“0”である場合（ステップS141：n，S143：n）、第一及び第二の出力ともに0が出力される（ステップS147）。

【0145】そして、それぞれの状態において、x，yのループが終わるまで上記処理が続き、それぞれf1(x, y) ～f4(x, y) を出力する（ステップS148～S151）。

【0146】即ち、上記のように加算累積のステップS301～304の後、1ブロック終了するまで次の画素の処理を継続する（ステップS305）。

【0147】図11及び図12に戻り、図8の実施例においての具体的な適応的加算／減算部50乃至53の動作について説明する。

【0148】それぞれの加算累積部30乃至33からのf1(x, y)～f4(x, y)出力は、対応する第一段目の適応的加算／減算部50、51及びオアゲート60、61に入力される。

【0149】適応的加算／減算部50について考察すると、S1、S2としてe1とe4が入力され、同時に第一、第二の入力として、f1(x, y)及びf4(x, y)が入力される。したがって、これらe1とe4によりf1(x, y)及びf4(x, y)の適応的加算/減算が行われ、g1(x,y)、g2(x,y)を出力する（ステップS501：図12）。

【0150】適応的加算／減算部51についても同様に処理が行われ、g3(x,y)、g4(x,y)を出力する（ステップS502）。

【0151】e1とe4は、オアゲート60に入力され、その論理和e5がオアゲート60から出力される。同様に、e2とe3は、オアゲート61に入力され、その論理和e6がオアゲート61から出力される（ステップS601）。

【0152】次いで、e5とe6をS1、S2として第二段目の適応的加算／減算部52、53に入力する。また、g1(x,y)とg3(x,y)が適応的加算／減算部52に、g2(x,y)とg4(x,y)が適応的加算／減算部53に入力する。

【0153】そして、適応的加算／減算部52において、e5とe6によるg1(x,y)とg3(x,y)の加算/減算が行われ、適応的加算／減算部53において、e5とe6によるg2(x,y)とg4(x,y)の加算/減算が行われ、第一の実施例の場合と同様にそれぞれf(x, y)を出力する（ステップS503、S504）。

【0154】尚、ステップS503、S504の加減算処理は、先に説明した式（5）～（9）にしたがって、アドレスを入れ換えて行われる。

【0155】図15は、本発明の方式と先に説明した既に提案した方式との演算回数の比較を示す図である。係数の個数を横軸として、縦軸に既に提案したテーブルを用いた二次元逆離散コサイン変換と本発明方式の加算回数を示す。

【0156】本発明の方式においては、条件判断により加算を適応的に行い、さらに、それぞれの係数の個数に対する最悪値（なるべく多くのC1～C4の集合に係数が割り振られた場合）を示している。この様に、全ての係数の個数において、本方式の方が、総演算回数が少ないことが分かる。

【0157】ここで、説明の簡単化のためIピクチャのみ復号できるものとし、M＝8に相当するMPEGを考える。上記実施例で用いたブロック図及び動作フローを用い、M＝8と置く。

【0158】更に、二次元逆離散コサイン変換は16ビット固定小数点で行うものとする。

【0159】かかる場合、4種類のテーブルf1'～f4'は、それぞれ以下のように構成される。（u，v）⊆Cnの係数に対して、数14のように表される。

【0160】

【数14】

$$fn'(level1, x, y, u, v) = \frac{1}{4}C(u)C(v)(level1*2+1)\cos\left(\frac{(2x+1)2}{16}\pi\right)\cos\left(\frac{(2y+1)2}{16}\pi\right)$$

ただし、0≦x，y＜8、－1024≦level1＜1023である

【0161】このテーブルサイズは、
1/2 ×(テーブルの種類) ×(uの種類) ×(vの種類) ×(xの種類) ×(yの種類) × (レベルの種類) =1/2 ×4 × (8／2)^4 ×2^{12} (ワード) ×2バイト=4194304 (バイト)
である。

【0162】これは、現在のパーソナルコンピュータで十分実現可能な大きさである。MPEGにおいては、逆量子化して得られる係数は奇数のみであるので、エントリとしては1以上の奇数のみを用意すればよい。本実施例では、
level1＝int（F（u，v）／2）
（但し、int（x）は小数点以下を切り捨てる。）として、level1を用いて、各カテゴリにおいてテーブルルックアップを行うことにより、逆DCTの計算がされる。

【0163】この場合の演算量は、係数が8個存在する場合、全ての演算量は192回であり、従来の方式に比べて、加算の演算量が大幅に削減されている。

【0164】更に、上記第一の実施例の方式を用いて、上記C1を4分割、C2、C3を2分割した場合、必要なテーブルは、2359296バイトとなり、更にテーブルサイズが小さくなり、その分演算量も削減される。

【0165】これまで説明した本発明の実施例では、係数の周波数成分と大きさの両方に関してテーブルを作成するものとして説明した。ここで、周波数成分が変わると、コサイン基底の乗算が必要で、浮動点少数演算が必要となるが、係数の大きさに関していえば、整数しか存

在しえない。

【0166】このようなことから、本発明の第三の実施例として、各集合のテーブルを作成する際に、周波数成分のみに関してテーブルを持ち、レベルF(u,v)に関しては、その値を乗ずることにより画素領域の値を求め、その後は、既に説明した手順に基づき逆ＤＣＴを行うことも可能である。

【0167】かかる本発明の第三の実施例について、以下詳細に説明する。

【0168】先ず第一の実施例と同様に、第三の実施例における二次元逆離散コサイン変換の手法について考察する。

【0169】上記数４から求められる集合Ｃ１～Ｃ４に属する各係数と０≦ｘ，ｙ＜Ｍ／２なるｘ，ｙに対して定義されるテーブルｆ1'～ｆ4'を以下の様に定義する。即ち、集合Ｃ１では、ｕ（＝２ｕ'）、ｖ（＝２ｖ'）、ｘ，ｙに対して数１５のように定義される。

【0170】

【数１５】

$$f1'(x,y,u,v)=\frac{1}{4}C(2u')C(2v')\cdot\cos\left(\frac{(2x+1)2u'}{2M}\pi\right)\cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

【0171】集合Ｃ２では、ｕ（＝２ｕ'＋１）、ｖ（＝２ｖ'）、ｘ，ｙに対して数１６のように定義される。

【0172】

【数１６】

$$f2'(x,y,u,v)=\frac{1}{4}C(2u'+1)C(2v')\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cos\left(\frac{(2y+1)2v'}{2M}\pi\right)$$

【0173】集合Ｃ３では、ｕ（＝２ｕ'）、ｖ（＝２ｖ'＋１）、ｘ，ｙに対して数１７のように定義される。

【0174】

【数１７】

$$f3'(x,y,u,v)=\frac{1}{4}C(2u')C(2v'+1)\cdot\cos\left(\frac{(2x+1)2u'}{2M}\pi\right)\cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

【0175】また集合Ｃ４では、ｕ（＝２ｕ'＋１）、ｖ（＝２ｖ'＋１）、ｘ，ｙに対して数１８のように定義される。

【0176】

【数１８】

$$f4'(x,y,u,v)=\frac{1}{4}C(2u'+1)C(2v'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cos\left(\frac{(2y+1)(2v'+1)}{2M}\pi\right)$$

【0177】そして、Ｆ（ｕ，ｖ）≠０　の条件を満たす（ｕ，ｖ）についてのみ、以下の数１９、数２０、数２１、数２２のように、集合Ｃ１～Ｃ４の中からこの係数の属するものを選択し、その領域に対して、係数の値を、先に数１５～数１８に示した、ｆ1'（ｘ，ｙ，ｕ，ｖ）～ｆ4'（ｘ，ｙ，ｕ，ｖ）のテーブルの値に乗じて加算を行って、ｆ１～ｆ４を求める。

【0178】ｆ１に対しては、（ｕ，ｖ）⊆Ｃ１の係数に対して、数１９となる。

【0179】

【数１９】

$$f1(x,y)=\sum_{(u,v),F(u,v)\neq 0}F(u,v)\cdot f1'(x,y,u,v)$$

【0180】ｆ２に対しては、（ｕ，ｖ）⊆Ｃ２の係数に対して、数２０となる。

【0181】

【数２０】

$$f2(x,y)=\sum_{(u,v),F(u,v)\neq 0}F(u,v)\cdot f2'(x,y,u,v)$$

【0182】ｆ３に対しては、（ｕ，ｖ）⊆Ｃ３の係数に対して、数２１となる。

【0183】

【数２１】

$$f3(x,y)=\sum_{(u,v),F(u,v)\neq 0}F(u,v)\cdot f3'(x,y,u,v)$$

【0184】ｆ４に対しては、（ｕ，ｖ）⊆Ｃ４の係数に対して、数２２となる。

【0185】

【数22】

$$f4(x,y) = \sum_{(u,v),F(u,v)\neq 0} F(u,v)\cdot f4'(x,y,u,v)$$

【0186】以上の処理を、全ての F（u，v）≠0 を満たす係数に対して行えば、f1～f4 が求まり、さらにf（x，y）が求まって、逆DCTが完了する。

【0187】ここで、予めf1'（x，y，u，v）～f4'（x，y，u，v）を計算して、テーブルとして持っておくことにより、f1～f4 に対しては、それぞれ集合C1～C4に属する係数が、F（u，v）≠0 を満たすものについてのみ、係数の値をf1'（x，y，u，v）～f4'（x，y，u，v）のテーブルの値に乗じて加算を行えば良いことになる。

【0188】f1～f4 が求まれば、式（1）～式（8）に従って画素データであるf（x，y）が求まる。

【0189】この第三の実施例の手法を用いてM×Mの二次元逆DCTを行う場合に必要なテーブルの数、および、演算量について考察を行う。

【0190】まず、数15～数18を、先に検討したレベルがLビットの場合の実現について考察する。ここでは、x，y，u，vともに、

0≦x，y，u，v＜M／2

を満たしている。

【0191】また、数15～数18に対して、それぞれ独立にテーブルが4種類必要である。これらのことより、数15～数18を実現するために必要なテーブルの大きさは、（テーブルの種類）×（uの種類）×（vの種類）×（xの種類）×（yの種類）＝4×（M／2）^{4} ＝M^4 ／4（ワード）

となる。M＝8で考えると高々1024ワードである。

【0192】次に、演算量について考察する。先の仮定と同様に、F（u，v）≠0 を満たす係数の個数をmとした場合について考える。ここで、0でないm個の係数を以下の様に分ける。

【0193】集合C1に属する係数の個数：m1

集合C2に属する係数の個数：m2

集合C3に属する係数の個数：m3

集合C4に属する係数の個数：m4

ここで、m1＋m2＋m3＋m4 ＝ m である。

【0194】したがって、加算の演算量については、先に説明したテーブルルックアップ方式と同様である。これにより、演算量も従来より削減される。

【0195】また、同様な計算で乗算量Jは、

J＝1/4 M^2 ×（m1＋m2＋m3＋m4）

＝1/4 M^2 ×m

である。

【0196】上記の第三の実施例の実現する構成ブロックの内、基本的概念ブロック構成は、第二の実施例の構成ブロック図である図8と同様に示されるので、再度の図示は、省略する。

【0197】更に、他の実施例と異なる特徴部分のブロック図は、図16に示され、更にその動作の特徴とする部分のフローチャートが図17、図18に示される。

【0198】他の実施例の場合と同様に実施例ブロック図を動作フローチャートと対照しながら説明する。

【0199】図8を参照すると、0係数判定部10及びカテゴリ判定部20の構成及び動作は、図2に示す場合と同様であり、0係数の判定（例えば、図11のステップS101参照）及びカテゴリの判定が行われる（例えば、図11のステップS201乃至S203参照）。

【0200】カテゴリ判定部20の出力は、対応する累積加算部30乃至33に入力され、累積加算される。第三の実施例における累積加算部30乃至33の構成例が図16に示される。

【0201】図17は、累積加算部30乃至33の対応する累積加算動作フローである。

【0202】図16において、F（u，v）として有効係数が一度でも入力される場合は、スイッチ307により“1”にセットする（ステップS181）。即ちスイッチ307により、“1”が出力されるように制御する。

【0203】テーブル308は、先に説明した実施例と異なり、レベルに関する情報を含まない周波数成分f'n(u,v,x,y) を登録記憶しており、集合Cn に属する(u,v) 及び(X,Y) により対応する画素領域の値が読みだされる。

【0204】この読みだされたf'n(u,v,x,y) と、F（u，v）が乗算器309により乗算され、対象性を省いた 1/2M× 1/2M の画素領域の値が求められる。

【0205】ついで、乗算器309の出力は、累積加算器303からの出力と加算器302において加算される（ステップS182）。これをx，yのループの終わりまで継続する（ステップS183）。

【0206】これにより、それぞれの係数が分類された集合ごとに、上記 1/2M× 1/2Mの画素領域の値を加算して累積される。

【0207】したがって、テーブル308は、F(u,v)により対応する画素領域の値として、周波数成分f'n(u,v,x,y) のみを記憶すればよくメモリ容量が削減出来る。

【0208】各累積加算部30乃至33から有効係数の有無により出力されるe1～e4 は、一段目の適応的加算／減算部50、51及びオアゲート回路60、61に入力される。

【0209】即ち、累積加算部30及び33からの出力e1 及びe4 は、適応的加算／減算部50及びオアゲー

ト回路６０に入力し、累積加算部３１及び３２からの出力e2 及びe3 は、適応的加算／減算部５１及びオアゲート回路６１に入力する。

【０２１０】一段目の適応的加算／減算部５０、５１及び、それらの出力及びオアゲート６０、６１の出力が入力される二段目の適応的加算／減算部５２、５３の構成例は、図１０に示されるものと同様である。

【０２１１】即ち、第一及び第二の入力Ｉｎｐ－１、Ｉｎｐ－２及び入力S1 （ｅｎ）とS2 （ｅｍ）の組み合わせに対応して、加算又は減算を行うように構成される演算回路である。また、図１８にその動作フローが示される。

【０２１２】図１８において、en 、em をそれぞれＳ１ 、S2 とすると、入力S1 とS2がともに“1”である場合（ステップＳ１９１：ｙ，Ｓ１９２：ｙ）、第一の出力として第一及び第二の入力Ｉｎｐ－１、Ｉｎｐ－２の和が、第二の出力として第一及び第二の入力Ｉｎｐ－１、Ｉｎｐ－２の差が演算され、出力される（ステップＳ１９４）。

【０２１３】更に、入力S1 （ｅｎ）のみが“1”である場合（ステップＳ１９１：ｙ，Ｓ１９２：ｎ）、第一及び第二の出力ともに第一の入力Ｉｎｐ－１が出力される（ステップＳ１９５）。

【０２１４】また、入力S1 （ｅｎ）が“0”であり、入力S2 （ｅｍ）が“1”である場合（ステップＳ１９１：ｎ，ステップＳ１９３：ｙ）、第一の出力にはＩｎｐ－２、第二の出力にはＩｎｐ－２に（－１）乗じたものが出力される（ステップＳ１９６）。

【０２１５】更に、入力S1 （ｅｎ）とS2 （ｅｍ）がともに“0”である場合（ステップＳ１９１：ｎ，Ｓ１９３：ｎ）、第一及び第二の出力ともに０が出力される（ステップＳ１９７）。

【０２１６】そして、それぞれの状態において、ｘ，ｙのループが終わるまで上記処理が続き、それぞれｆ1(x, y) ～ｆ4(x, y) を出力する（ステップＳ１９８～Ｓ２０１）。

【０２１７】次に、第三の実施例の拡張として、再帰的な手法について考察する。先ず、ここで、ｕ，ｖともに偶数成分である、カテゴリＣ１に属する係数に関する演算である上記数４の第一式、ｆ1(X, Y)を書き直すと、数２３のごとくなる。

【０２１８】

【数２３】

$$f1(x,y)=\sum_{u=0}^{M/2}\sum_{v=0}^{M/2}\tfrac{1}{4}F(2u',2v')C(2u')C(2v')\cos\left(\frac{(2x+1)u'}{2(M/2)}\pi\right)\cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right)$$

【０２１９】この数２３の式を良く見ると、次数がM／２の逆ＤＣＴの式になっていることが分かる。つまり、この式は、定係数以外は、数１のMをM／２に変え、ｕ（＝２ｕ’）をｕ’に、ｖ（＝２ｖ’）をｖ’に置き換えただけのものと同じ式である。

【０２２０】つまり、このｆ1 に関しては、

ｆ11（ｘ，ｙ）では、ｕ’，ｖ’ともに偶数（集合Ｃ１１）

ｆ12（ｘ，ｙ）では、ｕ’が奇数、ｖ’が偶数（集合Ｃ１２）

ｆ13（ｘ，ｙ）では、ｕ’が偶数、ｖ’が奇数（集合Ｃ１３）

ｆ14（ｘ，ｙ）では、ｕ’，ｖ’ともに奇数（集合Ｃ１４）

の様に数１を数４の各項に分解して、式（１）～式（８）の方法でｆ（ｘ，ｙ）を求めたのと全く同じ方法により、さらにｆ1 （ｘ，ｙ）に属する係数をｆ11～ｆ14のように４つの集合へ分割する。

【０２２１】これにより、ｆ1 （ｘ，ｙ）を求めるための演算量、テーブルサイズを小さくすることが可能である。

【０２２２】また、ｆ11（ｘ，ｙ）、ｆ12（ｘ，ｙ）、ｆ13（ｘ，ｙ）は、分割した基底の両方もしくは片方が偶成分であるため、さらなる分割も可能である。

【０２２３】次に、ｕが奇数成分、ｖが偶数成分である、集合Ｃ２に属する係数に関する演算である数４の第二項を書き直すと、数２４の如くになる。

【０２２４】

【数２４】

$$
\begin{aligned}
f2(x,y)
&= \sum_{u'=0}^{M/2} \sum_{v'=0}^{M/2} F(2u',2v') \cdot \tfrac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \\
&\quad \cdot C(2v') \cdot \cos\left(\frac{(2y+1)2v'}{2M}\pi\right) \\
&= \sum_{u'=0}^{M/2} \sum_{v'=0}^{M/2} F(2u',2v') \cdot \tfrac{1}{4} \cdot \underbrace{C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)}_{①} \\
&\quad \cdot \underbrace{C(2v') \cdot \cos\left(\frac{(2y+1)v'}{2(M/2)}\pi\right)}_{②}
\end{aligned}
$$

【０２２５】この数２４の式をみると、偶数成分であるｖに関しては②の様に次数M／２の１次元逆ＤＣＴの式となるが、奇数成分であるｕに関しては集合Ｃ１の場合と異なり、次数M／２の逆ＤＣＴの式とはならない。

【０２２６】この場合には、この式は以下の様に分解できる。ここでは、以下の様に、集合Ｃ２をＣ２１、Ｃ２２の２個の集合に分割する。

【０２２７】
ｈ２１（ｘ，ｙ）では、ｖ’が偶数（集合Ｃ２１）
ｈ２２（ｘ，ｙ）では、ｖ’が奇数（集合Ｃ２２）
更に、偶数のｖ’（⊇Ｃ２１）に対しては、下記数２５となる。

【０２２８】
【数２５】

$$
\begin{aligned}
h21(x,y) &= \\
&\sum_{u'=0}^{M/2} \sum_{v''=0}^{M/4} F(2u',4v'') \cdot \tfrac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \\
&\cdot C(4V'') \cdot \cos\left(\frac{(2y+1)2v''}{2(M/2)}\pi\right)
\end{aligned}
$$

【０２２９】また、奇数のｖ’（⊇Ｃ２２）に対しては、下記数２６となる。

【０２３０】
【数２６】

$$
\begin{aligned}
h22(x,y) &= \\
&\sum_{u'=0}^{M/2} \sum_{v''=0}^{M/4} F(2u',4v'') \cdot \tfrac{1}{4} C(2u'+1) \cdot \cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right) \cdot \\
&C(2(2v''+1)) \cdot \cos\left(\frac{(2y+1)(2v''+1)}{2(M/2)}\pi\right)
\end{aligned}
$$

【０２３１】そして、
（ｅ）０≦ｘ＜M／２，０≦ｙ＜M／４なる（ｘ，ｙ）に対して、
ｆ２（x，y）＝ｈ２１(X，y)＋ｈ２２(X，y)　・・・（９）
（ｆ）０≦ｘ＜M／２，M／４≦ｙ＜M／２なる（ｘ，ｙ）に対して、
ｆ２（x，y）＝ｈ２１(X，M/4−y)−ｈ２２(X，M/4−y)　・・・（１０）
なる式で表現できる。

【０２３２】したがって、この場合も、上記数１５〜数１８と同様に以下の様なテーブルを定義する。即ち、偶数のｖ’（＝２ｖ”）に対しては、下記数２７を定義する。

【０２３３】
【数２７】

$$h21'(x,y,u,v) =$$

$$\tfrac{1}{4}C(2u'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)\cdot C(4v'')\cdot\cos\left(\frac{(2y+1)2v''}{2(M/2)}\pi\right)$$

【０２３４】奇数のｖ’（＝２ｖ”＋１）に対しては、下記数２８を定義する。

【０２３５】

【数２８】

$$h22'(x,y,u,v) =$$

$$\tfrac{1}{4}C(2u'+1)\cdot\cos\left(\frac{(2x+1)(2u'+1)}{2M}\pi\right)$$

$$\cdot C(2(2v''+1))\cdot\cos\left(\frac{(2y+1)(2v''+1)}{2(M/2)}\pi\right)$$

【０２３６】そして、以下の式の様に、係数が属する領域に対して、係数の値を、先に示した数２７、数２８のｈ２１’(x,y,u,v)及びｈ２２’(x,y,u,v)のテーブルの値に乗じて加算を行い、ｈ２１～ｈ２２を求める。

【０２３７】ｈ２１に対しては、（ｕ，ｖ）⊆Ｃ２１の係数に対して、下記数２９になり、

【０２３８】

【数２９】

$$h21(x,y)=\sum_{(u,v),\,F(u,v)\neq 0} F(u,v)\cdot h21'(x,y,u,v)$$

【０２３９】ｈ２２に対しては、（ｕ，ｖ）⊆Ｃ２２の係数に対して、下記数３０となる。

【０２４０】

【数３０】

$$h22(x,y)=\sum_{(u,v),\,F(u,v)\neq 0} F(u,v)\cdot h22'(x,y,u,v)$$

【０２４１】この様に、テーブルと乗算を用いてｈ２１(x,y)，ｈ２２(x,y)を求める。次いで上記式（９）、（１０）を用いてｆ2(x,y)を求めることができる。この場合も、テーブルサイズ、演算量とも、上記数１６を用いて求めるより削減されている。

【０２４２】この場合、まずｈ２１(x,y)、ｈ２２(x,y)からｆ2(x,y)を求める。その後、式１～式８に示した方法でｆ(x,y)が求まる。

【０２４３】また、ｈ２１(x,y)に関しては、ｖ”成分が偶成分なので、さらなる分割が可能である。

【０２４４】以上に示した様に、その基底が偶函数の集合に対して、それらの基底は、次数を半分にしたＤＣＴの基底と等価となるので、その集合の更なる分割が可能となる。この様に、基底が偶函数の成分に関しては、更にその係数の集合を分割でき、その結果、演算量とテーブルサイズともにより小さくできる。

【０２４５】また、この分割は、必要な限り、もしくは基底が偶数であるかぎり再帰的に行うことが可能である。

【０２４６】上記第三の実施例の拡張の考え方において、更に、ｘ，ｙ，ｕ，ｖでテーブルを持つ代わりに、水平成分の（ｘ，ｕ）、垂直成分の（ｙ，ｖ）の組ごとにテーブルを独立に持ち（例えば、集合Ｃ２１、Ｃ２２においては、上記数２４の①と②）、それらのテーブルの値を乗じて加算すべき値を求めるようにすることが可能である。

【０２４７】この場合は、乗算回数は倍になるが、必要とするテーブルサイズが小さくなる。 更に、上記した予め、加算を行う際に片方の加算すべき集合に属する係数が存在するかどうかを元に条件判断し、適応的に加算を行うことにより、更に加算回数を削減する考えかたと同様に、まず、Ｎ個の各領域に有効係数があったかどうかのフラグを基に、それらの領域してひとつ広い領域の値を求める際に適応的に加算する。

【０２４８】そして、その新しい領域のフラグは、その領域を求めるのに用いた２つの領域の少なくとも何方か一方に有効係数があった場合に有効とする。この処理を、領域がＭ×Ｍになるまで行うようにすることが可能である。

【０２４９】次に、第四の実施例として上記第三の実施例において、テーブル３０８にｆ1’（ｘ，ｙ，ｕ，ｖ）～ｆ4’（ｘ，ｙ，ｕ，ｖ）のテーブルデータを持つ代わりに、以下の様なテーブルを定義することが可能である。即ち、０≦ｘ＜Ｍ／２、奇数のｕに対し、数３１を定義する。

【０２５０】

【数３１】

$$fe(x,u) = \tfrac{1}{2}C(u)\cos\left(\frac{(2x+1)u}{2M}\pi\right)$$

【０２５１】又、０≦ｘ＜Ｍ／２、偶数のｕに対し、数３２を定義する。

【０２５２】

【数３２】

$$fo(x,u) = \frac{1}{2}C(u)\cos\left(\frac{(2x+1)u}{2M}\pi\right)$$

【０２５３】したがって、f1 ～f4 は、以下の様にして求まる。

【０２５４】f1 に対しては、（ｕ，ｖ）⊆Ｃ１の係数に対して、数３３を定義する。

【０２５５】

【数３３】

$$f1(x,y) = \sum_{(u,v),\ F(u,v)\neq 0} F(u,v)\cdot fe(x,u)\cdot fe(y,v)$$

【０２５６】f2 に対しては、（ｕ，ｖ）⊆Ｃ２の係数に対して、数３４を定義する。

【０２５７】

【数３４】

$$f2(x,y) = \sum_{(u,v),\ F(u,v)\neq 0} F(u,v)\cdot fo(x,u)\cdot fe(y,v)$$

【０２５８】f3 に対しては、（ｕ，ｖ）⊆Ｃ３の係数に対して、数３５を定義する。

【０２５９】

【数３５】

$$f3(x,y) = \sum_{(u,v),\ F(u,v)\neq 0} F(u,v)\cdot fe(x,u)\cdot fo(y,v)$$

【０２６０】f4 に対しては、（ｕ，ｖ）⊆Ｃ４の係数に対して、数３６を定義する。

【０２６１】

【数３６】

$$f4(x,y) = \sum_{(u,v),\ F(u,v)\neq 0} F(u,v)\cdot fo(x,u)\cdot fo(y,v)$$

【０２６２】この後は、式（１）～式（８）と同じ方法でｆ（ｘ，ｙ）を同様に求まる。即ち、1/2 M×1/2 Mの画素ごとに、加算する画素において、その係数が表す縦周波数成分（ｕ）と横周波数成分（ｖ）と、その係数の大きさを乗算することにより求める方法である。

【０２６３】この第四の実施例方法での演算量は、第三の実施例に比べて加算の回数は変わらないものの、乗算の回数に関しては、２倍の　$J=M^2/2$　となる。

【０２６４】しかしながら、この手法のテーブルサイズは、

（ｘの種類）×（ｕの種類）×（テーブルの種類）

＝（Ｍ／２）×（Ｍ／２）×２

$=M^2/2$

となり、第三の実施例に比べて、テーブルサイズが大幅に削減される。この第四の実施例における加算累積部３０～３３の構成を図１９に示す。

【０２６５】図１９は、図１６との比較において、テーブル３０８が第一及び第二のテーブル３０８ａ、３０８ｂに分割されている。第一のテーブル３０８ａは、垂直周波数成分（ｕ）をテーブルとして記憶するものである。

【０２６６】第二のテーブル３０８ｂは、水平周波数成分（ｖ）をテーブルとして記憶するものである。したがって、更に乗算器３１０を設け、乗算器３０９と協動して、上記数３３～数３６の各々の乗算を行う。

【０２６７】更に、これら乗算の結果は、加算器３０２と、累積加算器３０３により累積さ上記数３３～数３６の各々の累積加算を行う。

【０２６８】更に、ＤＣＴ係数が集合Ｃ１～Ｃ４のうち、Ｃ１，Ｃ２，Ｃ４に属するもののみが存在し、Ｃ３に属するものが存在しない場合を考える。この場合、ｆ３（ｘ，ｙ）は０である。この場合、ｇ３（ｘ，ｙ）、ｇ４（ｘ，ｙ）は、単にｆ２（ｘ，ｙ）をコピーすれば良いだけで、加算が必要なくなる〔式（３）、（４）参照〕。

【０２６９】Ｃ３に加えて、Ｃ２に属する係数も存在しない場合は、もはや、ｇ３（ｘ，ｙ）、ｇ４（ｘ，ｙ）も０で、その結果ｇ３、ｇ４を求める為の加算は必要なく、さらに、ｆ（ｘ，ｙ）も単に、ｇ１（ｘ，ｙ）、ｇ２（ｘ，ｙ）のコピーだけで値が求まる。

【０２７０】このように、予め、加算を行う際に、片方の加算すべき集合に属する係数が存在するかどうかを基に条件判断し、適応的に加算を行うことにより、さらに平均的な加算回数を削減することが可能となる。

【０２７１】この片方の加算すべき集合に属する係数が存在するかどうかの判断は、０係数判定部１０で行われる。

【０２７２】次に、上記の第三の実施例以降の実施例において具体的な数字を当てはめて本発明の効果を考察する。ＭＰＥＧやＪＰＥＧ、Ｈ．２６１などの標準化で用いられているＭ＝８の場合を考える。

【０２７３】第三の実施例を考え、まずテーブルサイズを求める。１ワードあたり２バイトと仮定し、先の示した$M^4/4$（ワード）のテーブルサイズの大きさにＭ＝８を代入すると、２０４８バイト（２Ｋバイト）となる。インテル社製の８０４８６ＤＸ２のＭＰＵは８Ｋバ

イトのキャッシュを内蔵しているが、この大きさのテーブルなら全てがキャッシュ上に格納でき、メインメモリ上に大きなテーブルを専有することなく高速演算が期待できる。

【0274】また、1ブロックに8個の有効係数があるブロックの逆変換を考える。集合に属する係数毎に、対応する周波数の基本的な（つまり、係数の大きさ1に対する）1/2 M×1/2 M画素の大きさのパターンをテーブルとして持ち、対応する係数のパターンに係数の大きさを求める場合には、乗算回数は、128回、加算回数が256回となる。これは、明らかにＣｈｅｎの方式などより少ない。

【0275】また、第三の実施例の拡張手法を用いると、乗算回数が倍になり256回となる。この回数はＣｈｅｎの方式と同じであるが、ＣＰＵでこの方式を実現した場合、アルゴリズムが単純な分だけ、Ｃｈｅｎの方式よりも高速化が期待できる。またこの時のテーブルサイズは、先の場合と同様に1ワードが2バイトと仮定すれば、32バイトとなっており、必要なテーブルサイズがさらに小さくなる。

【0276】また、第四の実施例を用いて、Ｃ1をさらに4分割、Ｃ2、Ｃ3をそれぞれさらに2分割した場合、テーブルサイズは、1ワード2バイトとして、1152バイトとなる。

【0277】この場合、必要なテーブルサイズが先の2048バイトに比べてもおよそ半分になっている。またそれに応じてアルゴリズムは若干複雑になるものの、必要な演算回数もより少なくすることが可能である。

【0278】次に本発明の第五の実施例として、複数の二次元逆離散コサイン変換のうち最適なものを選んで、最悪値見積もりが可能な二次元逆離散コサイン変換を実現する方式を説明する。

【0279】上記実施例に従い説明したように、本発明の改良型テーブルルックアップ方式により有効係数が少ない時には、従来から知られているどの方式よりも高速に二次元逆離散コサイン変換を行うことが可能である。

【0280】既に検討したように、特に係数の数が17より少ないときは、Ｃｈｅｎの手法よりも乗算回数を削減することができ、また係数の数が9より少ない時は、現在知られているなかで最も乗算回数が少ないＨａｑｕｅの手法により、乗算回数を削減することができる。

【0281】しかしながら、本発明の手法では、係数の数が多いときには、従来の高速演算より逆に演算量が増加してしまう。また、演算速度が係数の数に左右されてしまうため、演算速度の最悪値の見積もりが困難である。

【0282】したがって、本発明の第五の実施例は、かかる問題を解決するものである。

【0283】図20は、第五の実施例の基本概念を説明するブロック図である。従来の高速手法は、係数の数によらず演算速度が一定である。また、上記本発明の実施例では、係数が少ないときは高速であるものの、係数の個数が多いときには演算量が増大してしまう。

【0284】そのため、予めブロックの有効係数の数を数え、その個数の時に、最も高速な手法を選ぶことにより、係数が多いブロックに対しても、最悪値を極端に増大させることなく逆離散コサイン変換を可能とする。

【0285】図20において、210は、有効係数の数を数えるカウンタであり、220は、カウンタ210からの有効係数の数を入力し、離散コサイン変換を行う方式を選択する離散コサイン変換選択部である。

【0286】上記のように各方式を採用する場合において、有利な有効係数の数が求められるので、離散コサイン変換選択部220に有利な有効係数の数と対応づけて記憶して置く。

【0287】231～23ｎは、複数の離散コサイン変換方式の各々に対応し、演算処理を行う演算回路である。離散コサイン変換選択部220は、カウンタ210から入力される有効係数の数を参照して、有効な離散コサイン変換を行う方式を選択して切替え信号を入力スイッチＳＷ1、出力スイッチＳＷ2に送る。

【0288】したがって、入力スイッチＳＷ1、出力スイッチＳＷ2により切替え接続された演算回路231～23ｎのいずれかに有効係数が入力される。

【0289】このように、予めブロックの有効係数の数をカウントし、その個数に対応する最も有利な方式を選択することにより、係数の多いブロックに対しても、最悪値を極端に増大させることなく逆離散コサイン変換を実行することが可能である。

【0290】図21は、図20に対応する第五の実施例ブロック図であり、8×8の逆離散コサイン変換を行う回路である。図21においては，本発明とＣｈｅｎの方法を切り換える実施例である。

【0291】Ｍ＝8として、8×8のブロックがカウンタ210に入力する。カウンタ210は、有効係数をカウントし、カウントした有効係数の個数を離散コサイン変換選択部220に入力する。

【0292】離散コサイン変換選択部220には、有効係数の個数ｎとして、ｎ＞16の時、Ｃｈｅｎ方式を対応づけ、ｎ≦16の時、本発明の演算手法に対応付けるべくテーブルが記憶されている。

【0293】したがって、演算回路として本発明の改良型テーブルルックアップ方式による演算回路231と、Ｃｈｅｎのアルゴリズムによる演算回路232を有する。

【0294】スイッチＳＷ1、ＳＷ2により有効係数カウンタ210において、有効係数の個数ｎとして、ｎ＞16の時、演算回路232に、ｎ≦16の時、演算回路231に切り換えられる。これにより、全体として、高速な二次元逆離散コサイン変換が実現される。

【０２９５】

【発明の効果】本発明によれば、二次元離散コサイン変換とその逆変換を用いたシステムにおいて、小さなテーブルサイズで復号処理速度を高速化することができる。特にマイクロプロセッサを用いて構成された、画像復号器に適用した場合に効果が大きい。

【０２９６】また、上記の通り実施例にしたがい本発明を説明したが、かかる実施例は、本発明の説明のためのものであり、本発明は、かかる実施例に限定されるものではない。

【０２９７】本発明の技術思想と一致するもの及び均等の範囲にあるものは、本発明の保護の範囲に含まれるものである。

【図面の簡単な説明】

【図１】Ｉピクチャ用ＭＰＥＧデコーダの構成図である。

【図２】本発明の第一の実施例ブロック図である。

【図３】図２におけるカテゴリ判定部の構成例である。

【図４】図２における加算累積部の構成例である。

【図５】図２における加算／減算部の構成例である。

【図６】第一の実施例動作フロー（その１）である。

【図７】第一の実施例動作フロー（その２）である。

【図８】本発明の第二の実施例ブロック図である。

【図９】図８における加算累積部の構成例である。

【図１０】図８における加算／減算部の構成例である。

【図１１】第二の実施例動作フロー（その１）である。

【図１２】第二の実施例動作フロー（その２）である。

【図１３】第二の実施例動作フロー（その３）である。

【図１４】第二の実施例動作フロー（その４）である。

【図１５】本発明の効果を説明する図である。

【図１６】本発明の第三の実施例における加算累積部の構成例である。

【図１７】図１６における加算累積部の動作フローである。

【図１８】本発明の第三の実施例の拡張における適応的加算／減算のフローチャートである。

【図１９】本発明の第四の実施例における加算累積部の構成例である。

【図２０】本発明の第五の実施例の原理ブロック図である。

【図２１】本発明の第五の実施例の具体的実施例である。

【図２２】先に提案された逆ＤＣＴ方式のブロック図である。

【図２３】先に提案された逆ＤＣＴ方式の動作フローである。

【符号の説明】

１　可変復号化回路

２　逆量子化回路

３　二次元逆離散コサイン変換回路

１０　０係数判定部

２０　カテゴリ判定部

３１～３２　加算累積部

４０～４３　加算／減算部

５０～５３　適応的加算／減算部

６０、６１　オアゲート

３０１、３０５、３０８、３０８ａ、３０８ｂ　ＩＤＣＴ係数テーブル

３０２、４０１　加算回路

３０３　累積加算メモリ

４０２　減算回路

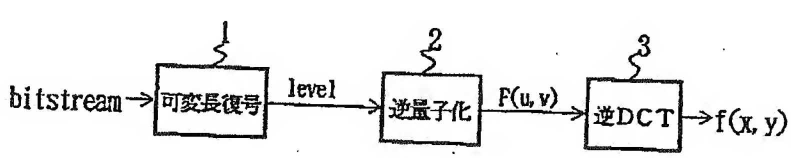
３０４　絶対値化回路

３０６　加算／減算回路

３０９、３１０　乗算器

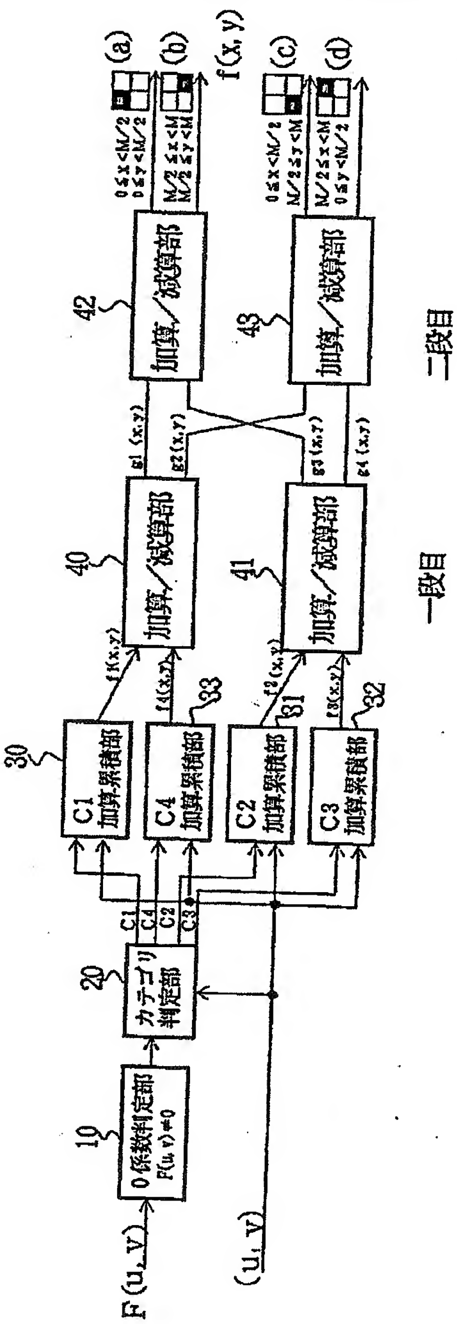
【図１】

Ｉピクチャ用ＭＰＥＧデコーダの構成図



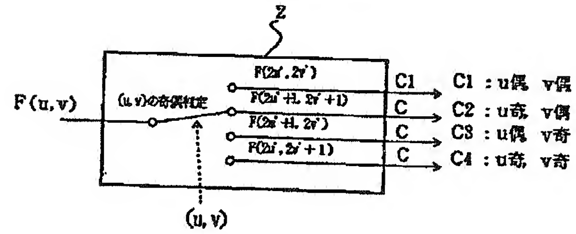
【図2】

本発明の第一実施例ブロック図



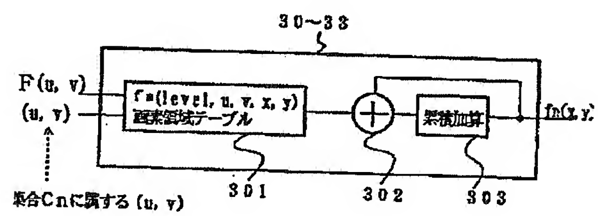
【図3】

カテゴリ判定部の構成例



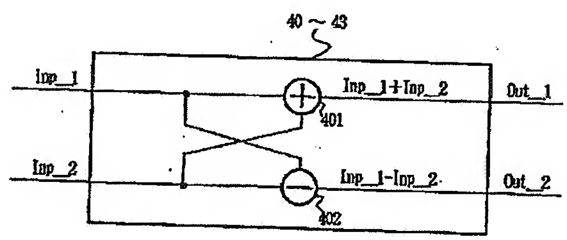
【図4】

加算累積部の構成例



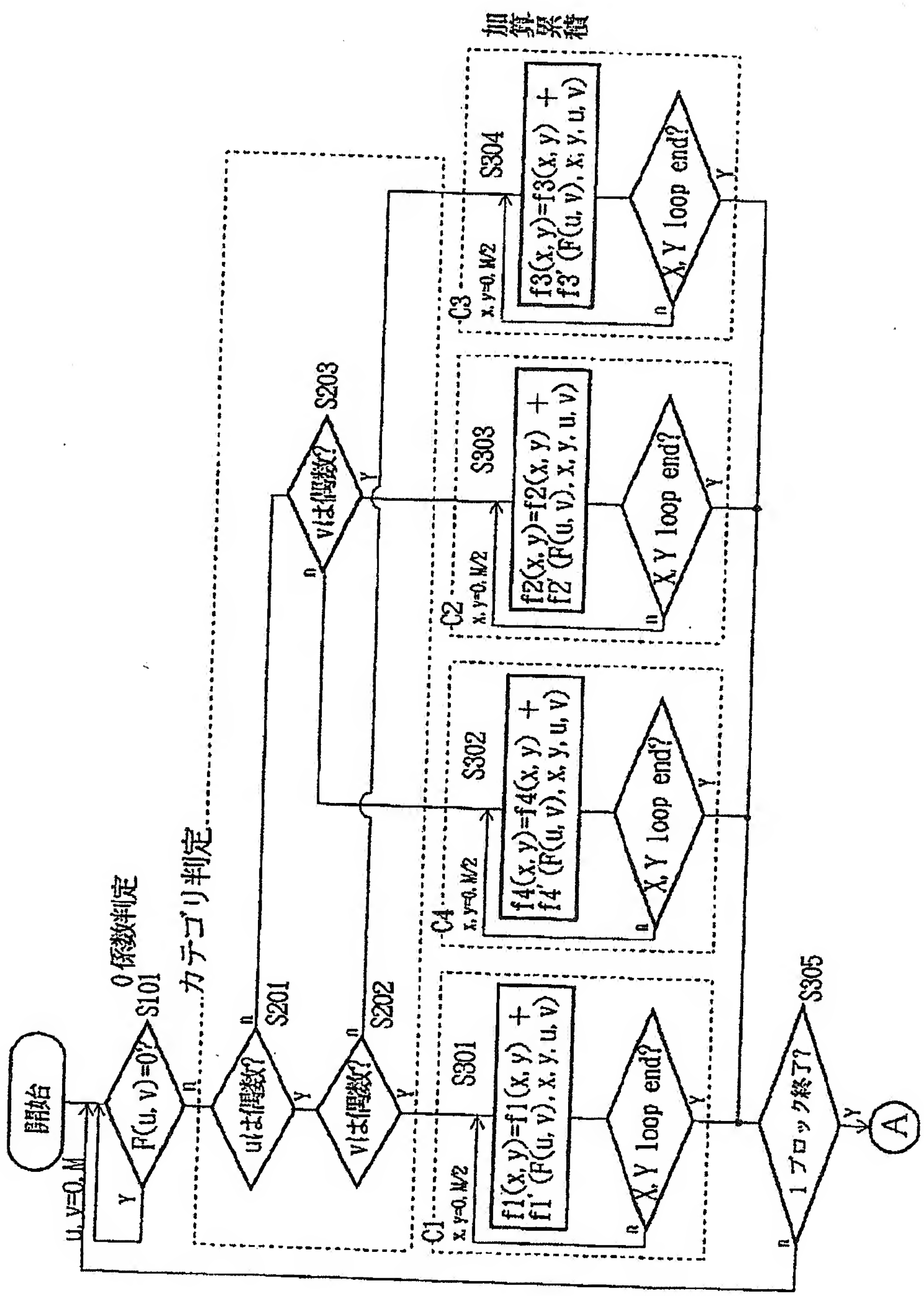
【図5】

加算／減算部の構成例



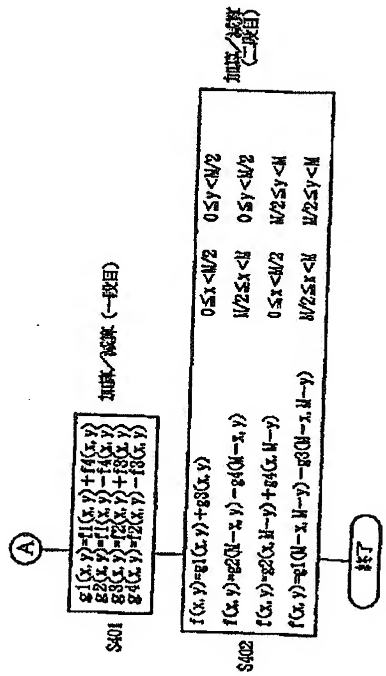
【図6】

第一の実施例動作フロー（その1）



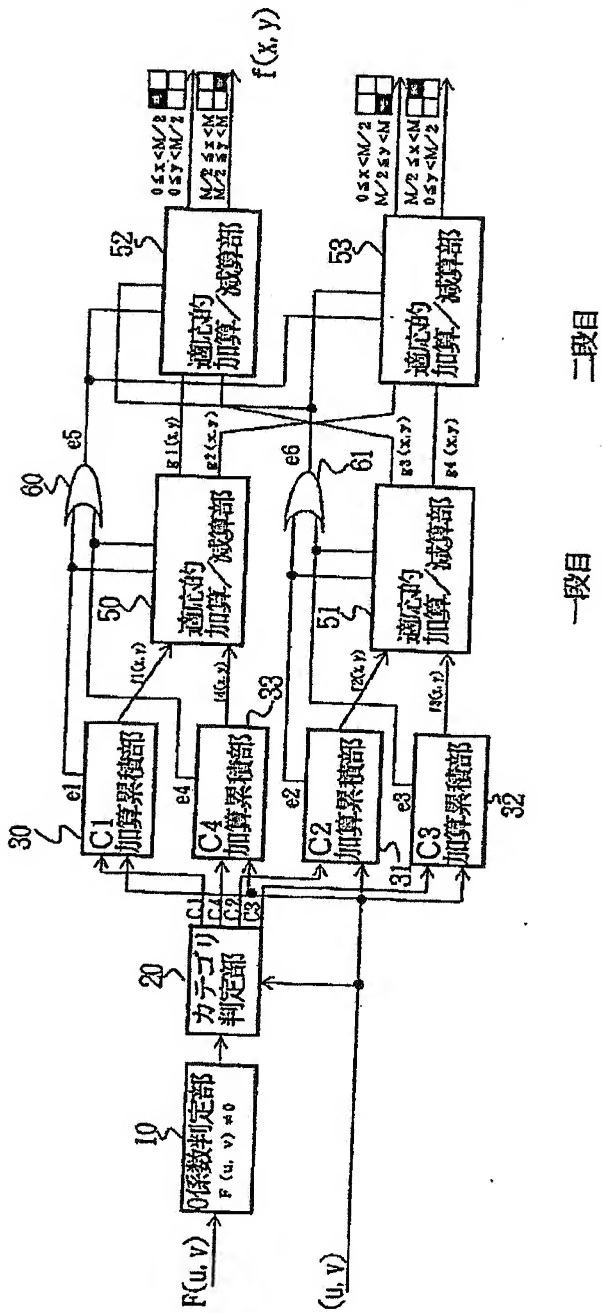
【図7】

第一の実施例動作フロー（その2）



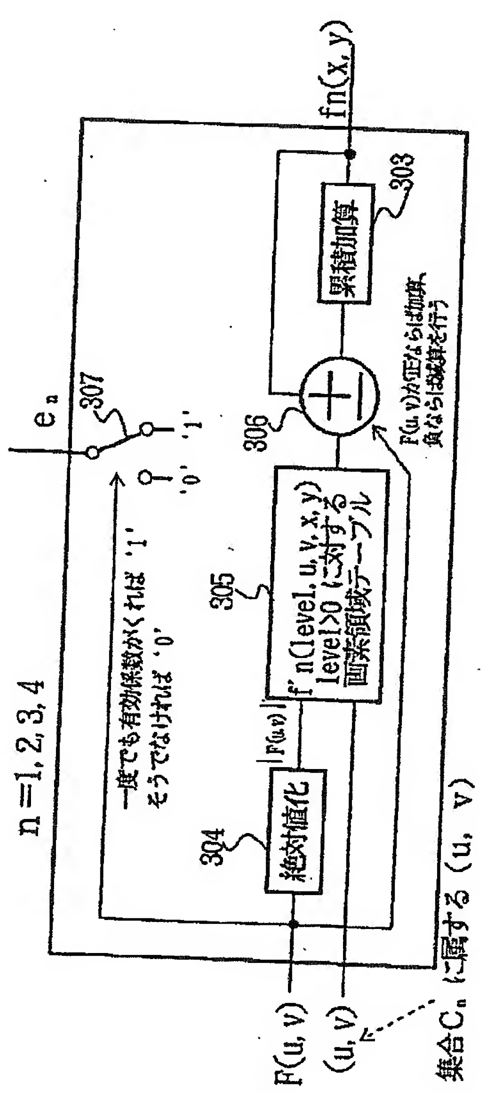
【図8】

本発明の第二の実施例ブロック図



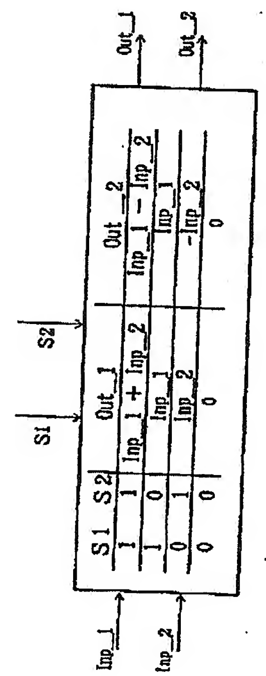
【図9】

加算累積部の構成例



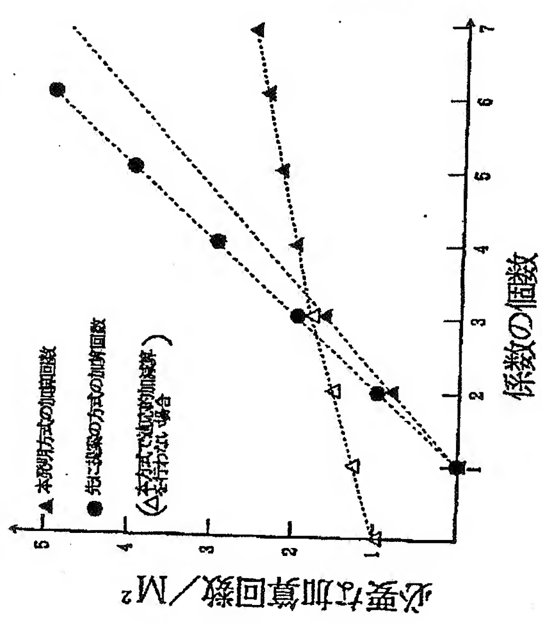
【図10】

加算／減算部の構成例



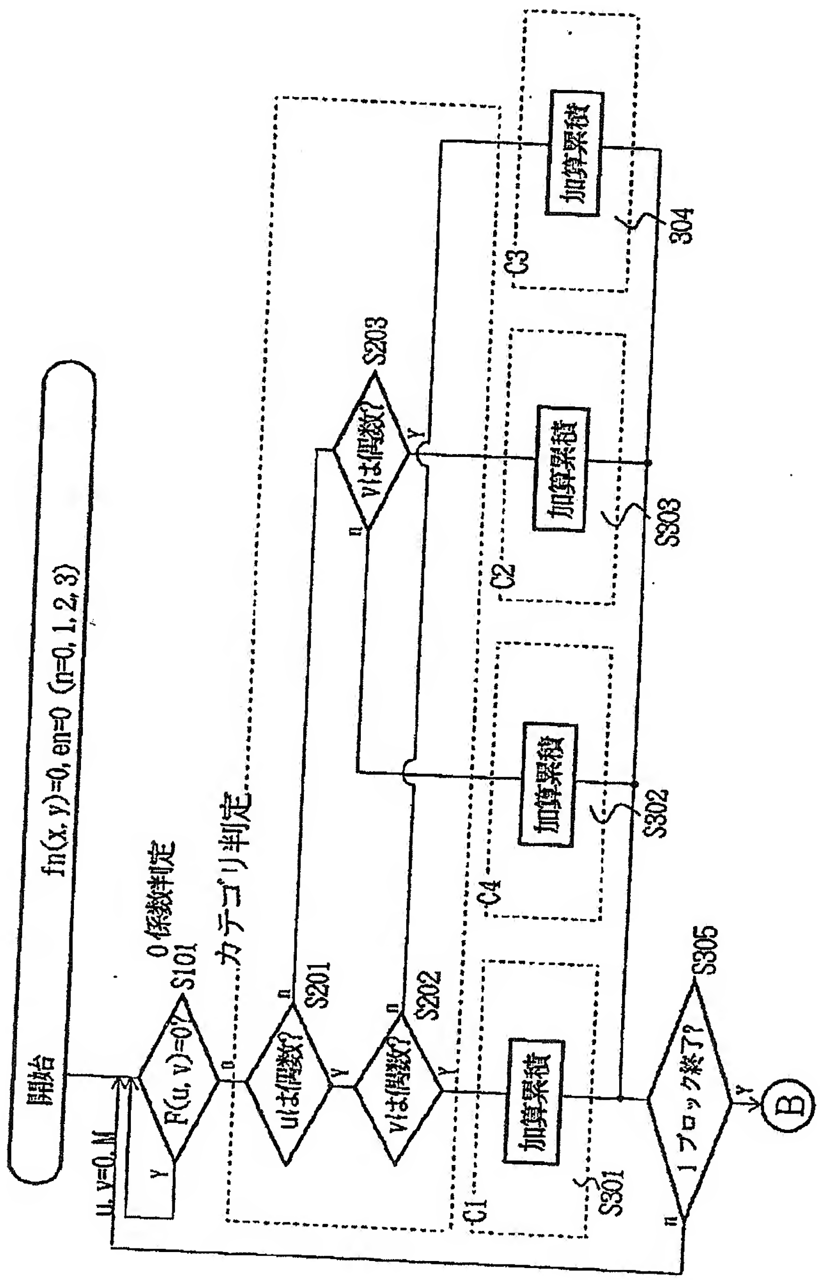
【図15】

本発明の効果を説明する図



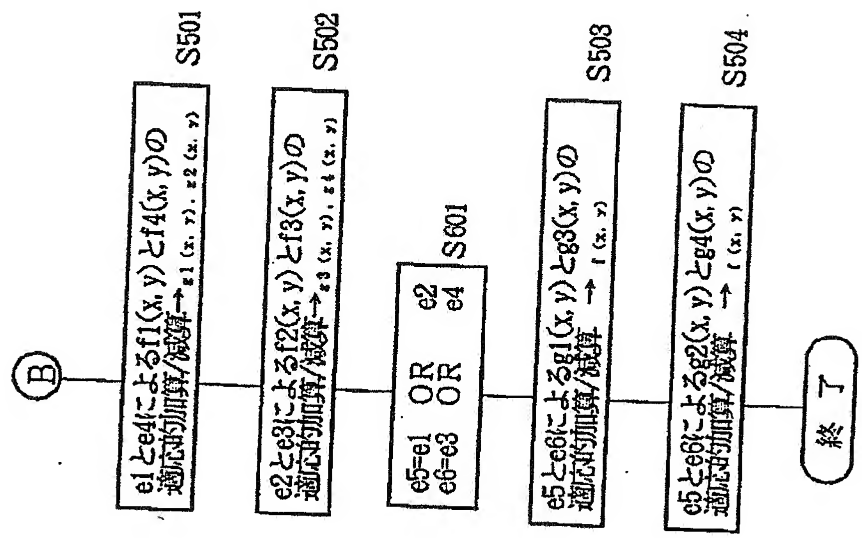
【図１１】

第二の実施例動作フロー（その１）



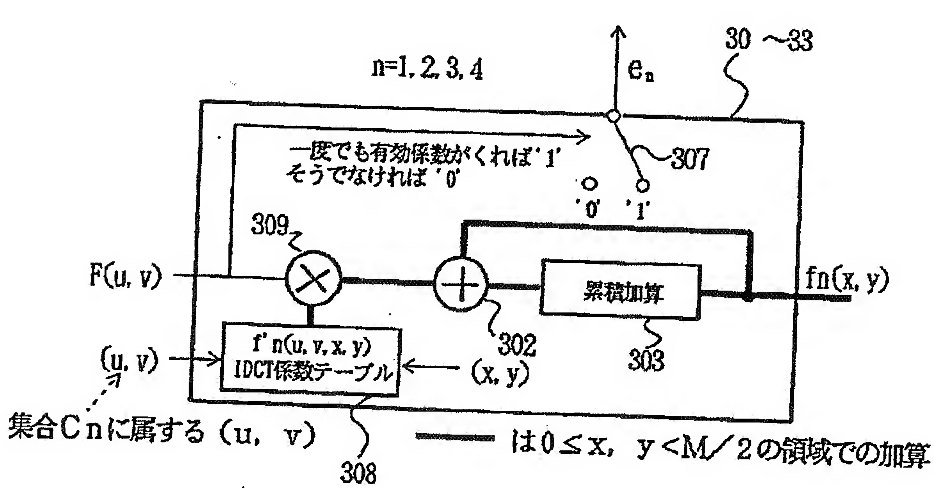
【図12】

第二の実施例動作フロー（その2）

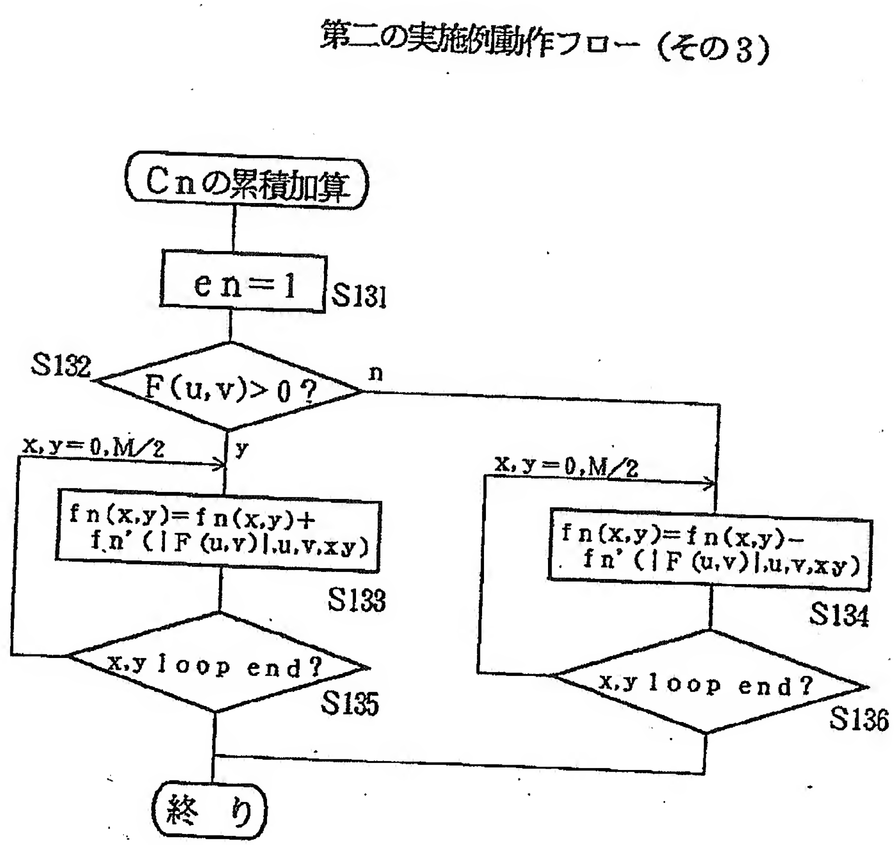


【図16】

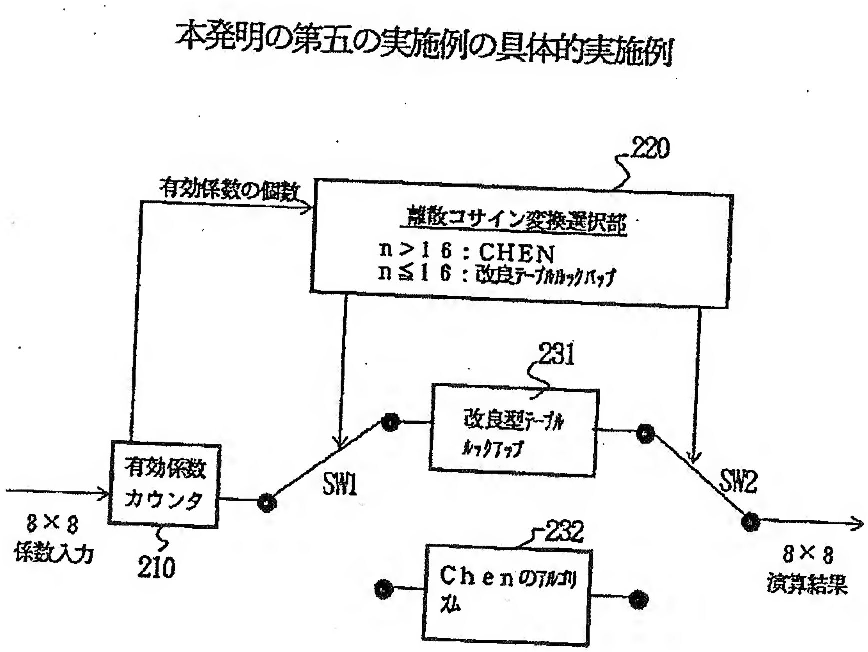
加算累積部の構成例



【図13】

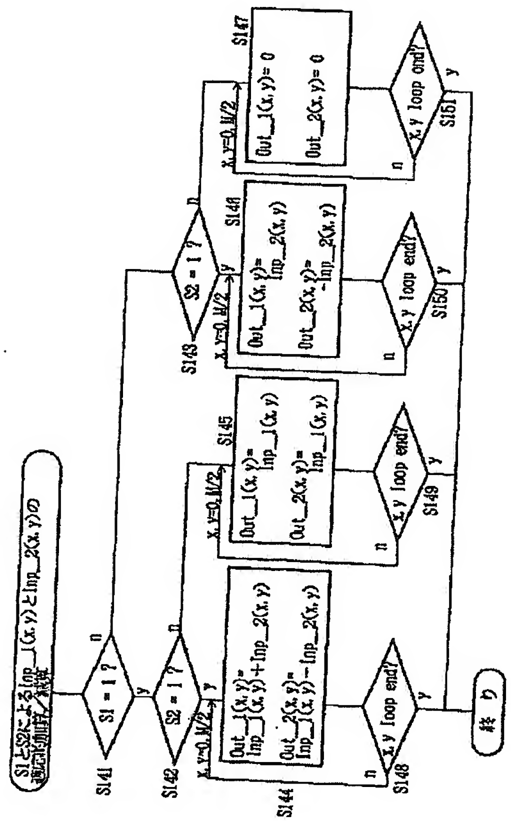

第二の実施例動作フロー（その3）
Cnの累積加算
en=1
S131
S132
F(u,v)>0？
n
y
x,y=0,M/2
fn(x,y)=fn(x,y)+
fn'(|F(u,v)|,u,v,x,y)
S133
x,y loop end？
S135
x,y=0,M/2
fn(x,y)=fn(x,y)−
fn'(|F(u,v)|,u,v,x,y)
S134
x,y loop end？
S136
終り

【図21】


本発明の第五の実施例の具体的実施例
220
有効係数の個数
離散コサイン変換選択部
n＞16：CHEN
n≦16：改良テーブルルックアップ
231
改良型テーブルルックアップ
有効係数カウンタ
SW1
SW2
8×8
係数入力
210
232
Chenのアルゴリズム
8×8
演算結果

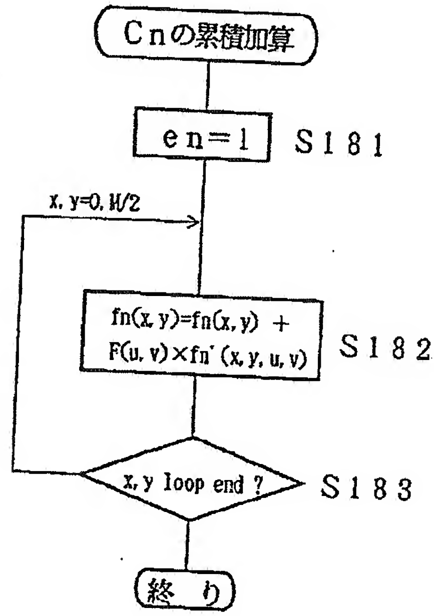
【図１４】

第二の実施例（その４）



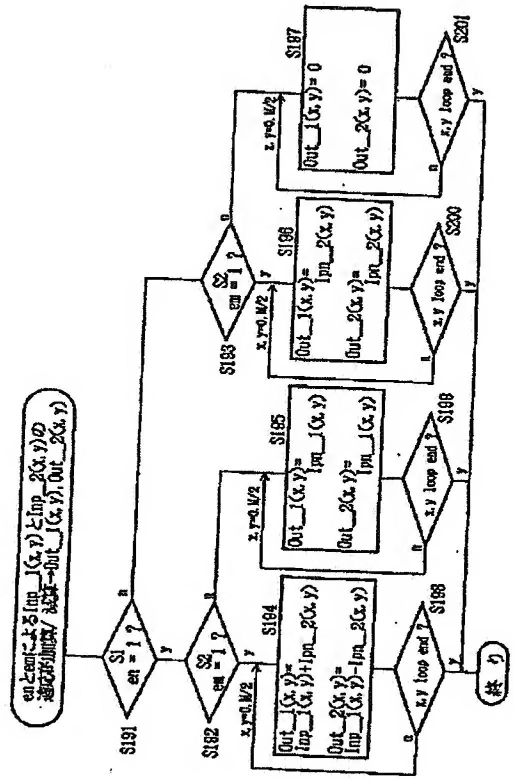
【図１７】

加算累積のフローチャート



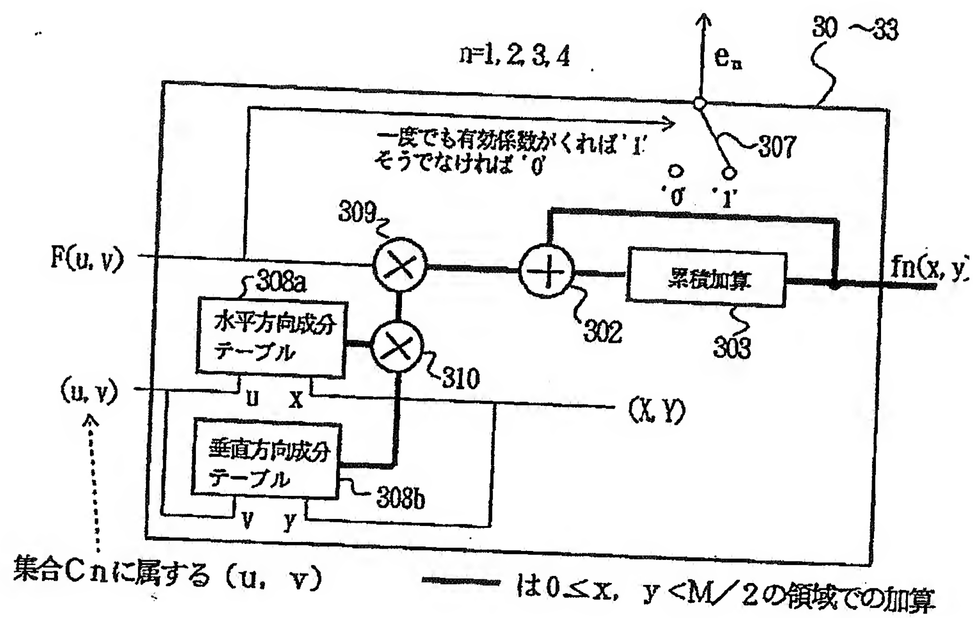
【図18】

適応的加算／減算のフローチャート



【図１９】

加算累積部の構成例



【図２０】

本発明の第五の実施例の原理ブロック図